

CFM 2409 US



日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年12月27日

出 願 番 号

Application Number:

特願2000-399331

出 願 人

Applicant(s):

キヤノン株式会社

RECEIVED

MAR 14 2002

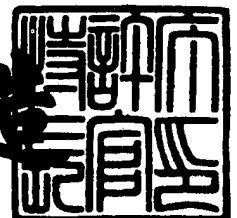
Technology Center 2100

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年11月16日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3100529

【書類名】 特許願

【整理番号】 4367014

【提出日】 平成12年12月27日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 1/00

【発明の名称】 データ処理装置及びその制御方法、データ処理方法、コンピュータ可読メモリ

【請求項の数】 11

【発明者】

 【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

 【氏名】 中山 忠義

【特許出願人】

 【識別番号】 000001007

 【氏名又は名称】 キヤノン株式会社

【代理人】

 【識別番号】 100076428

 【弁理士】

 【氏名又は名称】 大塚 康德

 【電話番号】 03-5276-3241

【選任した代理人】

 【識別番号】 100101306

 【弁理士】

 【氏名又は名称】 丸山 幸雄

 【電話番号】 03-5276-3241

【選任した代理人】

 【識別番号】 100115071

 【弁理士】

 【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241

【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0001010

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ処理装置及びその制御方法、データ処理方法、コンピュータ可読メモリ

【特許請求の範囲】

【請求項 1】 データを処理するデータ処理装置であって、

データを保持する保持手段と、複数の加算器もしくは減算器もしくは加減算器のいずれかと、更に乗算器もしくは位取り変換器とを有する演算ユニットとを複数個備え、

前記複数の演算ユニットは、 n 個従属に接続した構成からなり、入力する $2n + 1$ タップと $2n - 1$ タップのデータから、2 種類のウェーブレット変換係数の演算を行う

ことを特徴とするデータ処理装置。

【請求項 2】 前記演算ユニットは、前記乗算器もしくは位取り変換器の出力を、前記保持手段の入力もしくは出力データのどちらか一方、あるいは入力・出力データの両方に加算もしくは減算できるように、該保持手段の前後に 2 つの加算器もしくは減算器もしくは加減算器を有する

ことを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 3】 前記演算ユニットは、前記位取り変換器を 2 つ有し、前記演算に応じて 2 種類の位取りを切り換える

ことを特徴とする請求項 2 に記載のデータ処理装置。

【請求項 4】 前記演算ユニットは、該演算ユニットへの入力データの 1 つに前記保持手段の出力データを加算すると共に、前記位取り変換器を複数有し、前記演算に応じて 2 種類の位取りを切り換える

ことを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 5】 前記演算ユニットは、前記加算器もしくは減算器にオフセットを入力するオフセット生成手段と、該演算ユニット間に配置された該演算ユニットの出力データの丸め処理を行う丸め処理手段を有する

ことを特徴とする請求項 3 あるいは請求項 4 に記載のデータ処理装置。

【請求項 6】 前記オフセット生成手段と丸め処理手段それぞれにおいて、

データをマスクする第 1 マスク手段と第 2 マスク手段を有し、第 1 マスク手段と第 2 マスク手段とが排他的に有効となるように制御する

ことを特徴とする請求項 5 に記載のデータ処理装置。

【請求項 7】 前記演算ユニットは、前記保持手段を少なくとも 2 つ有し、前記 2 つの保持手段それぞれに 2 ラインのデータをライン交互に入力して、前記演算を行う

ことを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 8】 データを処理するデータ処理装置の制御方法であって、データを保持する保持手段と、複数の加算器もしくは減算器もしくは加減算器のいずれかと、更に乗算器もしくは位取り変換器とを有する演算ユニットとが n 個従属に接続されて構成された演算ユニット群の先頭の演算ユニットへ、データを入力する入力工程と、

入力した $2n+1$ タップと $2n-1$ タップのデータから、ウェーブレット変換係数の演算を行うために、前記演算ユニット群を制御する制御工程とを備えることを特徴とするデータ処理装置の制御方法。

【請求項 9】 データを処理するデータ処理方法であって、データを入力する入力工程と、前記入力工程で入力されたデータに基づいて、L i f t i n g 格子構造上の格子点データを演算する途中のデータである中間データを生成する第 1 生成工程と

前記生成工程で生成された中間データを記憶媒体に保持する保持工程と、前記保持工程で保持された中間データを用いて、前記格子点データを生成する第 2 生成工程と

を備えること特徴とするデータ処理方法。

【請求項 10】 データを処理するデータ処理装置の制御のプログラムコードが格納されたコンピュータ可読メモリであって、

データを保持する保持手段と、複数の加算器もしくは減算器もしくは加減算器のいずれかと、更に乗算器もしくは位取り変換器とを有する演算ユニットとが n 個従属に接続されて構成された演算ユニット群の先頭の演算ユニットへ、データ

を入力する入力工程のプログラムコードと、

入力した $2n+1$ タップと $2n-1$ タップのデータから、ウェーブレット変換係数の演算を行うために、前記演算ユニット群を制御する制御工程のプログラムコードと

を備えることを特徴とするコンピュータ可読メモリ。

【請求項 11】 データを処理するデータ処理のプログラムコードが格納されたコンピュータ可読メモリであって、

データを入力する入力工程のプログラムコードと、

前記入力工程で入力されたデータに基づいて、L i f t i n g 格子構造上の格子点データを演算する途中のデータである中間データを生成する第 1 生成工程のプログラムコードと、

前記生成工程で生成された中間データを記憶媒体に保持する保持工程のプログラムコードと、

前記保持工程で保持された中間データを用いて、前記格子点データを生成する第 2 生成工程のプログラムコードと

を備えることコンピュータ可読メモリ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データを処理するデータ処理装置及びその制御方法、データ処理方法、コンピュータ可読メモリに関するものである。

【0002】

【従来の技術】

画像、特に、多値画像は非常に多くの情報を含んでおり、その画像を蓄積・伝送する際にはデータ量が膨大になってしまうという問題がある。このため画像の蓄積・伝送に際しては、画像の持つ冗長性を除く、あるいは画質の劣化が視覚的に認識し難い程度で画像データの変化を許容することによってデータ量を大幅に削減する高能率符号化が用いられる。

【0003】

例えば、静止画像の国際標準符号化方式として I S O と I T U - T により勧告された J P E G では、画像データをブロック（8 画素×8 画素）ごとに離散コサイン変換（D C T）した後に、変換によって得られる各変換係数を各々量子化し、さらにエントロピー符号化することにより画像データを圧縮している。この圧縮では、画像データをブロックごとに D C T、量子化を行なっているため、復号画像の各ブロックの境界で、所謂ブロック歪みが見える場合がある。

【 0 0 0 4 】

一方、新しい静止画像の国際標準符号化方式として検討されている J P E G 2 0 0 0 では、量子化・エントロピー符号化の前に行なう変換処理として、ウェーブレット変換が用いられている。ウェーブレット変換は、D C T 変換のように小さなブロック単位で処理を行うのではなく、該ブロックより十分大きなサイズのタイル単位で変換処理をするため、前記ブロック歪みがないといった特徴がある。

【 0 0 0 5 】

以下では、L i f t i n g S c h e m e を用いたウェーブレット変換フィルタ処理について説明する。

【 0 0 0 6 】

J P E G 2 0 0 0 で使われているウェーブレット変換は、L i f t i n g S c h e m e という方法で処理をすると、少ない演算量で効率良く変換処理を行なうことができる。

【 0 0 0 7 】

図 1 に順方向の L i f t i n g S c h e m e、図 2 に逆方向の L i f t i n g S c h e m e のシグナルフローを示す。図 1、図 2 は、低域ウェーブレット変換係数の演算に 9 タップのデータ、高域ウェーブレット変換係数の演算に 7 タップのデータを用いる場合のシグナルフローである。図の中の α 、 β 、 γ 、 δ は L i f t i n g 係数と呼ばれるものである。

【 0 0 0 8 】

以下、図 1 の動作について説明する。

【 0 0 0 9 】

入力画素を $X_0, X_1, X_2, X_3, X_4, X_5, \dots, X_n$ のように順に表わす。
 該入力画素は、分類ユニット 2 0 1 にて、偶数画素系列と奇数画素系列とに分類され、該分類ユニット 2 0 1 の上側には添字が偶数の画素 $X_0, X_2, X_4, \dots, X_{2n}$ が、下側には添字が奇数の画素 $X_1, X_3, X_5, \dots, X_{2n+1}$ が出力される。

初段の *L i f t i n g* 処理では、偶数画素系列に対し *L i f t i n g* 係数： α を乗算し、連続する 2 個の乗算結果を、該 2 画素の中央に位置する奇数画素系列中の画素に加算する。

【0 0 1 0】

これを一般化した式で表現すると、以下のようになる。

【0 0 1 1】

$$D_{2n+1} = X_{2n+1} + \alpha \cdot X_{2n} + \alpha \cdot X_{2n+2} \quad (1)$$

2 段目の *L i f t i n g* 処理では、新たに得られた奇数画素系列 $D_1, D_3, D_5, \dots, D_{2n+1}$ に対し *L i f t i n g* 係数： β を乗算し、連続する 2 個の乗算結果を、該 2 画素の中央に位置する偶数画素系列中の画素に加算する。

【0 0 1 2】

これを一般化した式で表現すると、以下のようになる。

【0 0 1 3】

$$E_{2n+2} = X_{2n+2} + \beta \cdot D_{2n+1} + \beta \cdot D_{2n+3} \quad (2)$$

3 段目の *L i f t i n g* 処理では、*L i f t i n g* 係数： γ を用いて、初段と同じように、4 段目の *L i f t i n g* 処理では、*L i f t i n g* 係数： δ を用いて、2 段目と同じように処理する。3 段、4 段目の *L i f t i n g* 処理内容を一般化した式で表現すると、以下のようになる。

【0 0 1 4】

$$H_{2n+1} = D_{2n+1} + \gamma \cdot E_{2n} + \gamma \cdot E_{2n+2} \quad (3)$$

$$L_{2n+2} = E_{2n+2} + \delta \cdot H_{2n+1} + \delta \cdot H_{2n+3} \quad (4)$$

図中、 K は、低域・高域ウェーブレット係数を正規化するものであるが、本発明の本質を説明するにあたって、特に関係ないことであるので、以下、説明を省略する。

【0015】

正規化処理を無視すれば、3段、4段目のLifting処理によって得られる H_n 、 L_n は各々ウェーブレット高域変換係数とウェーブレット低域変換係数に対応する。

【0016】

次に、図2に示す逆方向のLifting Schemeのシグナルフローについて簡単に説明する。まず始めに、順方向のLifting Schemeにおける正規化処理に対応して、逆の係数を掛けた後、4段のLifting処理を行なう。各段の処理内容を一般化した式で表現すると、以下のようになる。

$$(1 \text{ 段目}) \quad E_{2n+2} = L_{2n+2} - \delta \cdot H_{2n+1} - \delta \cdot H_{2n+3} \quad (5)$$

$$(2 \text{ 段目}) \quad D_{2n+1} = H_{2n+1} - \gamma \cdot E_{2n} - \gamma \cdot E_{2n+2} \quad (6)$$

$$(3 \text{ 段目}) \quad X_{2n+2} = E_{2n+2} - \beta \cdot D_{2n+1} - \beta \cdot D_{2n+3} \quad (7)$$

$$(4 \text{ 段目}) \quad X_{2n+1} = D_{2n+1} - \alpha \cdot X_{2n} - \alpha \cdot X_{2n+2} \quad (8)$$

上記(5)～(8)式は、各々(4)～(1)式を移項して得られるものである。

【0017】

以上が、Lifting Schemeを用いたウェーブレット変換フィルタ処理についての説明であり、次に、該Lifting Schemeに再帰的演算を組み合わせたウェーブレット変換フィルタ処理について説明する。

【0018】

図1及び図2のLifting Schemeを別の視点から表現したものが、図3及び図4に示すLifting格子構造である。同図において、□は入力データを、○は格子点（あるいは格子点データ演算器）を表わし、○から出ている矢印は格子点データの流れを示す。これらの図はLifting Schemeにおける基本処理（前記(1)～(8)式の処理）並びに該処理によって得られる新たなデータを1つの格子点に対応させたものである。

【0019】

図3に示す順方向変換のLifting格子構造では、1つの格子点データは前記(1)～(4)式のいずれかを用いて計算される。

【 0 0 2 0 】

図 4 に示す逆方向変換の L i f t i n g 格子構造では、1 つの格子点データは前記 (5) ～ (8) 式のいずれかにより計算される。

【 0 0 2 1 】

L i f t i n g 格子構造を見ることによって、入力データや各格子点データの依存関係が一目瞭然となる。例えば、変換出力データである L 4 を計算するには、9 つの入力データ：X 0 ～ X 8 が必要であるが、3 つの格子点データ：H 3、E 4、H 5 だけからも計算できることが分かる。

【 0 0 2 2 】

L 4、H 5 を計算し出力した時点で、X 8、D 7、E 6、H 5 の 4 つのデータを残しておくことが可能である。この 4 つのデータと新たな入力データ X 9、X 10 とから、D 9、E 8、H 7、L 6 の順に 4 つの格子点データを計算することができる。そして、L 6 と H 7 を変換データとして出力すると共に、X 10、D 9、E 8、H 7 の 4 つのデータを残しておいて次の計算に利用するといった効率的な演算処理が可能である。

【 0 0 2 3 】

ところで、この演算処理を効率的に行うには、先頭から偶数添え字の画素データと奇数添え字の画素データのペアを 1 つの単位にして処理する必要がある。上記の説明では、入力する画素データのペアが奇数添え字から始まるため、先頭の偶数添え字の画素データが半端になってしまう。

【 0 0 2 4 】

水平方向のウェーブレット変換処理の場合、半端となるのは先頭 1 画素だけであるが、垂直方向のウェーブレット変換処理では、先頭 1 ラインのデータが半端なデータになってしまう。その上、大部分の画像データの垂直サイズは偶数なので、最後の 1 ラインも半端になってしまう。

【 0 0 2 5 】

ハードウェアで処理した場合、ペアの 2 ラインを処理する時間と先頭の 1 ラインのみを処理する時間は同じであるので、画像の先頭と最後で生ずる半端な 1 ラインは処理効率が悪い。

【 0 0 2 6 】

【発明が解決しようとする課題】

上述したように、L i f t i n g 処理の処理結果を保存しておき、それを再利用することによって効率的な演算処理をしようすると、データストリームの途中のデータは2ライン同時に処理ができるが、偶数サイズの画像では先頭と最後のラインのみ1ライン単独で処理しなければならず効率が良くない。

【 0 0 2 7 】

本発明は上記の課題を解決するためになされたものであり、効率的にウェーブレット変換処理を実行できるデータ処理装置及びその制御方法、データ処理方法、コンピュータ可読メモリを提供することを目的とする。

【 0 0 2 8 】

【課題を解決するための手段】

上記の目的を達成するための本発明によるデータ処理装置は以下の構成を備える。即ち、

データを処理するデータ処理装置であって、

データを保持する保持手段と、複数の加算器もしくは減算器もしくは加減算器のいずれかと、更に乗算器もしくは位取り変換器とを有する演算ユニットとを複数個備え、

前記複数の演算ユニットは、 n 個従属に接続した構成からなり、入力する $2n + 1$ タップと $2n - 1$ タップのデータから、2種類のウェーブレット変換係数の演算を行う。

【 0 0 2 9 】

また、好ましくは、前記演算ユニットは、前記乗算器もしくは位取り変換器の出力を、前記保持手段の入力もしくは出力データのどちらか一方、あるいは入力・出力データの両方に加算もしくは減算できるように、該保持手段の前後に2つの加算器もしくは減算器もしくは加減算器を有する。

【 0 0 3 0 】

また、好ましくは、前記演算ユニットは、前記位取り変換器を2つ有し、前記演算に応じて2種類の位取りを切り換える。

【 0 0 3 1 】

また、好ましくは、前記演算ユニットは、該演算ユニットへの入力データの1つに前記保持手段の出力データを加算すると共に、前記位取り変換器を複数有し、前記演算に応じて2種類の位取りを切り換える。

【 0 0 3 2 】

また、好ましくは、前記演算ユニットは、前記加算器もしくは減算器にオフセットを入力するオフセット生成手段と、該演算ユニット間に配置された該演算ユニットの出力データの丸め処理を行う丸め処理手段を有する。

【 0 0 3 3 】

また、好ましくは、前記オフセット生成手段と丸め処理手段それぞれにおいて、データをマスクする第1マスク手段と第2マスク手段を有し、第1マスク手段と第2マスク手段とが排他的に有効となるように制御する。

【 0 0 3 4 】

また、好ましくは、前記演算ユニットは、前記保持手段を少なくとも2つ有し

前記2つの保持手段それぞれに2ラインのデータをライン交互に入力して、前記演算を行う。

【 0 0 3 5 】

上記の目的を達成するための本発明によるデータ処理装置の制御方法は以下の構成を備える。即ち、

データを処理するデータ処理装置の制御方法であって、

データを保持する保持手段と、複数の加算器もしくは減算器もしくは加減算器のいずれかと、更に乗算器もしくは位取り変換器とを有する演算ユニットとが n 個従属に接続されて構成された演算ユニット群の先頭の演算ユニットへ、 $2n+1$ タップと $2n-1$ タップのデータを入力する入力工程と、

前記入力した $2n+1$ タップと $2n-1$ タップのデータによって、ウェーブレット変換係数の演算を行うために、前記演算ユニット群を制御する制御工程とを備える。

【 0 0 3 6 】

上記の目的を達成するための本発明によるデータ処理方法は以下の構成を備える。即ち、

データを処理するデータ処理方法であって、

データを入力する入力工程と、

前記入力工程で入力されたデータに基づいて、L i f t i n g 格子構造上の格子点データを演算する途中のデータである中間データを生成する第 1 生成工程と

前記生成工程で生成された中間データを記憶媒体に保持する保持工程と、

前記保持工程で保持された中間データを用いて、前記格子点データを生成する第 2 生成工程と

を備える。

【 0 0 3 7 】

上記の目的を達成するための本発明によるコンピュータ可読メモリは以下の構成を備える。即ち、

データを処理するデータ処理装置の制御のプログラムコードが格納されたコンピュータ可読メモリであって、

データを保持する保持手段と、複数の加算器もしくは減算器もしくは加減算器のいずれかと、更に乗算器もしくは位取り変換器とを有する演算ユニットとが n 個従属に接続されて構成された演算ユニット群の先頭の演算ユニットへ、データを入力する入力工程のプログラムコードと、

前記入力した $2n+1$ タップと $2n-1$ タップのデータから、ウェーブレット変換係数の演算を行うために、前記演算ユニット群を制御する制御工程のプログラムコードと

を備えることを特徴とするコンピュータ可読メモリ。

【 0 0 3 8 】

上記の目的を達成するための本発明によるコンピュータ可読メモリは以下の構成を備える。即ち、

データを処理するデータ処理のプログラムコードが格納されたコンピュータ可読メモリであって、

データを入力する入力工程のプログラムコードと、

前記入力工程で入力されたデータに基づいて、L i f t i n g 格子構造上の格子点データを演算する途中のデータである中間データを生成する第 1 生成工程のプログラムコードと、

前記生成工程で生成された中間データを記憶媒体に保持する保持工程のプログラムコードと、

前記保持工程で保持された中間データを用いて、前記格子点データを生成する第 2 生成工程のプログラムコードと

を備える。

【 0 0 3 9 】

【発明の実施の形態】

以下、図面を参照して本発明の好適な実施形態を詳細に説明する。

<実施形態 1>

本発明では、従来例で述べた「L 4、H 5 を計算し出力した時点で X 8，D 7，E 6，H 5 の 4 つのデータを残しておき、新たなデータ X 9 と X 1 0 を入力する」といった処理を次のように変える。

【 0 0 4 0 】

すなわち、図 3 において「L 4、H 5 を計算し出力した時点で D 9、E 8、H 7、L 6 の 4 つのデータを演算する途中のデータ（以下、中間データとも称する）D 9 t、E 8 t、H 7 t、L 6 t を残しておき、次の処理サイクルで新たなデータ X 1 0 と X 1 1 を入力する」といった処理を行う。

【 0 0 4 1 】

尚、実施形態 1 では、低域・高域ウェーブレット変換係数の演算にそれぞれ、9 タップと 7 タップのデータを用いる場合、つまり、 9×7 フィルタ（9 タップと 7 タップのデータからなるフィルタ）の場合について説明するが、これに限定されず、本発明は、低域・高域ウェーブレット変換係数の演算にそれぞれ、 $2n + 1$ タップと $2n - 1$ タップのデータに対して適用できる。

【 0 0 4 2 】

まず、上述の中間データは、以下のように示される。

【 0 0 4 3 】

$$D 9 t = X 9 + \alpha \cdot X 8 \quad (9)$$

$$E 8 t = X 8 + \beta \cdot D 7 \quad (10)$$

$$H 7 t = D 7 + \gamma \cdot E 6 \quad (11)$$

$$L 6 t = E 6 + \delta \cdot H 5 \quad (12)$$

同時に、前の処理サイクルで残しておいた $D 7 t$ 、 $E 6 t$ 、 $H 5 t$ 、 $L 4 t$ から、 $L 4$ 、 $H 5$ を以下の演算で計算する。

【 0 0 4 4 】

$$D 7 = D 7 t + \alpha \cdot X 8 \quad (13)$$

$$E 6 = E 6 t + \beta \cdot D 7 \quad (14)$$

$$H 5 = H 5 t + \gamma \cdot E 6 \quad (15)$$

$$L 4 = L 4 t + \delta \cdot H 5 \quad (16)$$

残しておくデータはそれぞれレジスタで保持しておいて次の処理サイクルで利用する。該データを用いて次の処理サイクルで行なう演算は以下の内容である。

【 0 0 4 5 】

$$D 9 = D 9 t + \alpha \cdot X 10 \quad (17)$$

$$E 8 = E 8 t + \beta \cdot D 9 \quad (18)$$

$$H 7 = H 7 t + \gamma \cdot E 8 \quad (19)$$

$$L 6 = L 6 t + \delta \cdot H 7 \quad (20)$$

上記処理によって、 $L 6$ 、 $H 7$ を出力することができる。

【 0 0 4 6 】

この処理内容を *L i f t i n g* 格子構造の図にならって表現したのが図 5 である。実際のハードウェア構成を図 6 に示す。同図において、

6 0 1、6 0 3 は、上記ペアデータを入力する端子である。6 0 5、6 0 7 は、それぞれ低域ウェーブレット変換係数と高域ウェーブレット変換係数を出力する端子である。6 1 1 は、*L i f t i n g* 係数を乗算係数として乗算する乗算器である。6 1 3 は、演算途中のデータである中間データを保持するレジスタである。6 1 5、6 1 7 は、乗算器の出力をレジスタの入力と出力データに加算する加算器である。6 2 1 は、*L i f t i n g* 演算ユニットである。6 2 2、6 2 3

、 6 2 4 は、 L i f t i n g 演算ユニット 6 2 1 とは乗算器の乗算係数が違うだけで他はすべて同じ構成の L i f t i n g 演算ユニットである。

【 0 0 4 7 】

入力端子 6 0 1 と 6 0 3 に、 X 8 と X 9 を入力している時、乗算器 6 1 1 にて、 X 8 に L i f t i n g 係数 α が乗算され、乗算結果として $\alpha \cdot X 8$ が出力される。この乗算結果は、それぞれ加算器 6 1 5、 6 1 7 に送られ、加算器 6 1 5 では前記 (9) 式、加算器 6 1 7 では、前記 (1 3) 式が演算される。

【 0 0 4 8 】

他の L i f t i n g 演算ユニット 6 2 2、 6 2 3、 6 2 4 においても下記に示す対応関係で演算処理が行なわれる。すなわち、

L i f t i n g 演算ユニット 6 2 2 では、前記 (1 0) 式と (1 4) 式

L i f t i n g 演算ユニット 6 2 3 では、前記 (1 1) 式と (1 5) 式

L i f t i n g 演算ユニット 6 2 4 では、前記 (1 2) 式と (1 6) 式の演算が同時に行なわれる。

【 0 0 4 9 】

よって、 L i f t i n g 演算ユニット 6 2 4 では低域ウェーブレット変換係数 L 4 が、 L i f t i n g 演算ユニット 6 2 3 では高域ウェーブレット変換係数 H 5 が演算され、それらの係数は端子 6 0 5、 6 0 7 から出力される。(9) ~ (1 2) 式の演算結果は、各演算ユニット内のレジスタに保持され、次のサイクルで実行される (1 7) ~ (2 0) 式の演算に利用される。

【 0 0 5 0 】

以上のようにして、水平方向の順方向のウェーブレット変換処理を行なうことができる。

【 0 0 5 1 】

本発明人は、図 7 に示す格子点演算ユニットを多段接続してウェーブレット変換処理を行なうデータ処理装置について既に提案している。この提案内容と本発明の内容の違いは以下の 2 点である。

【 0 0 5 2 】

(i) 格子点演算ユニットと L i f t i n g 演算ユニットの内部構成の違い

(ii) 同時に処理する 2 つのデータの組み合わせの違い（入力位相の違いとも言える）

その他については基本的に同じである。特に、それらの演算ユニットを多段接続して処理する接続方法に関してはまったく同じであり、それらの演算ユニットをブラックボックス化して、入力データの位相を無視すれば、2 つの処理系はまったく同じに見える。

【 0 0 5 3 】

よって、前記格子点データ演算ユニットを *L i f t i n g* 演算ユニットに置き換えるだけで、既に提案した内容における応用例は、本発明に対してそのまま適用できる。

【 0 0 5 4 】

応用例 1 水平方向の逆方向ウェーブレット変換処理装置

応用例 2 垂直方向の順方向ウェーブレット変換処理装置

応用例 3 垂直方向の逆方向ウェーブレット変換処理装置

応用例 4 異なる種類のデータを多重化処理可能なウェーブレット変換処理装置

応用例 5 順方向と逆方向の切換可能なウェーブレット変換処理装置（水平・垂直）

応用例 6 上記応用例 5 において係数正規化用の乗算器を共用したウェーブレット変換処理装置

上記各応用例について簡単に説明する。

【 0 0 5 5 】

応用例 1 は図 8 に示す構成で実現される。図 6 の構成との違いは、*L i f t i n g* 係数の順序が上から δ 、 γ 、 β 、 α となっており、図 6 に対して上下逆の順序になっている。また、図 6 におけるレジスタ前後の加算器は、図 8 ですべて減算器に置き換わっている。（係数として $-\delta$ 、 $-\gamma$ 、 $-\beta$ 、 $-\alpha$ を用いるのなら加算器のままでもよい。）

応用例 2 と 3 は、それぞれ図 6 と図 8 の各演算ユニット内のレジスタをすべてラインメモリに置き換えるだけでよい。（図面は省略する。）

応用例 4 は、図 6 の各演算ユニット内のレジスタを、1 個から複数個の従属接続に変更し、複数種類のデータを多重化してから、端子 6 0 1, 6 0 3 へ入力することで実現できる。

【 0 0 5 6 】

応用例 5 は、図 6 の各演算ユニット内の加算器をすべて加減算器に置き換え、順方向変換時には加算器として使用し、逆方向変換時には減算器として使用すると共に、演算ユニット間にセレクタを配して、処理したデータが下の演算ユニットからすぐ上の演算ユニットへ流れるように制御し、図 8 の構成と等価になるようにしたものである。該構成を図 9 に示す。

【 0 0 5 7 】

応用例 6 は応用例 5 に対し、さらにもう 1 つセレクタを設け、係数正規化用の乗算器を順方向変換時と逆方向変換時で共用できるようにしたものである。該構成を図 1 0 に示す。

【 0 0 5 8 】

7 番目の応用例として、

応用例 7 画像境界部のデータを拡張することなく境界処理が可能なウェーブレット変換処理装置である。

【 0 0 5 9 】

境界処理を一言で説明すると、「格子点データの演算に必要な 3 つのデータの内、2 つのデータしか存在しない場合、足りないデータを対称位置にあるデータで置き換えて演算する」といった処理になる。

【 0 0 6 0 】

この場合、本発明における *L i f t i n g* 演算ユニットの内部構成の変更を必要とする。

【 0 0 6 1 】

図 1 1、1 2、1 3 に前記変更後の内部構成を示す。変更後の構成は以下に対応する 3 種類となる。

【 0 0 6 2 】

構成 1 始端境界処理が可能な L i f t i n g 演算ユニット（始端対応演算ユニット）

構成 2 終端境界処理が可能な L i f t i n g 演算ユニット（終端対応演算ユニット）

構成 3 始端境界処理と終端境界処理の両方が可能な L i f t i n g 演算ユニット（両端対応演算ユニット）

図 1 1 に示す始端境界処理が可能な L i f t i n g 演算ユニットは、図 1 4 に示す左端境界部分の●で示した格子点データを演算することができる。図 1 1 で追加された 2 つのセレクタ 1 1 2 1 と 1 1 2 3 は通常処理では端子 a を選択する。尚、図 1 1 ～図 1 3 に示す C は、L i f t i n g 係数： α 、 β 、 γ 、 δ のいずれかである。

【 0 0 6 3 】

境界部のデータを演算する場合、まず、セレクタ 1 1 2 1 で端子 b を選択することで、端子 1 1 0 3 から入力するデータをそのままの値でレジスタ 1 1 1 3 に格納する。次のサイクルで、セレクタ 1 1 2 3 の端子 b を選択することによって、端子 1 1 0 1 から入力したデータを $2 \cdot C$ 倍した値を前記レジスタの出力に加算する。

【 0 0 6 4 】

例えば、端子 1 1 0 3 から入力したデータ X_0 がレジスタ 1 1 1 3 にそのままの値で格納され、次のサイクルで、端子 1 1 0 1 に D_1 が入力されると $X_0 + 2 \cdot \beta \cdot D_1$ ($= E_0$) が端子 1 1 3 1 から出力される。これらの制御は、セレクタ 1 1 2 1、1 1 2 3 に与える制御信号で行なう。

【 0 0 6 5 】

図 1 2 に示す終端境界処理が可能な L i f t i n g 演算ユニットは、図 1 4 に示す右端境界部分の●で示した格子点データを演算することができる。前記始端境界処理と異なり、終端境界処理ではレジスタ 1 2 1 3 への格納時に、端子 1 2 0 1 から入力したデータを $2 \cdot C$ 倍した値を端子 1 2 0 3 から入力したデータに加算するよう制御する。その代わり、該レジスタ 1 2 1 3 の出力へは何も加算しない。

【0066】

上記 *L i f t i n g* 演算ユニットを図16のように接続した構成で、図14に示す *L i f t i n g* 格子構造の格子点データをすべて演算することができる。*L i f t i n g* 格子構造は図14だけではなく、図15のような場合もある。図15の境界データを処理するには、図16の構成において、始端対応演算ユニットと終端対応演算ユニットを入れ替えればよい。しかし、図14と図15の両方を処理できるようにするには、図13に示す構成の両端対応演算ユニットを図17のように従属接続した構成が必須となる。

【0067】

順方向と逆方向の両方のウェーブレット変換が可能な *L i f t i n g* 演算ユニットとして、図18に示す構成も考えられる。これは、図6に示した演算ユニット621に乗算器1853を付加し、セレクタ1855で、元からある乗算器1851の出力から追加した乗算器1853の出力へ切り換えることで、逆方向のウェーブレット変換に使用できる *L i f t i n g* 演算ユニットとなる。

【0068】

つまり、*L i f t i n g* 演算ユニットは、乗算係数 (*L i f t i n g* 係数) が α である乗算器1851を乗算係数 (*L i f t i n g* 係数) が $-\delta$ の乗算器1853へ切り換え、同様にして、図6に示した演算ユニット622～623それぞれに対応する不図示の *L i f t i n g* 演算ユニットにおいては、 β は $-\gamma$ に、 γ は $-\beta$ に、 δ は $-\alpha$ に切り換えることで、図6の構成は図8の構成と等価になるため、逆方向のウェーブレット変換ができるようになる。

【0069】

上記のような乗算係数の異なる乗算器を2個使用する構成は、図7に示す演算ユニットにも適用でき、図19に示す格子点データ演算ユニットとなる。該演算ユニットを用いた場合にも、順方向と逆方向の両方のウェーブレット変換を行なうことができる。

【0070】

以上説明したように、実施形態1によれば、*L i f t i n g* 格子構造上の格子点データを演算する過程で生成される中間データをレジスタに保存し、この中間

データを用いた演算を可能にすることで、入力されたデータストリームを先頭から2画素ずつペアにしてウェーブレット変換処理を行うことができる。また、簡単な回路構成で、画像境界部のデータを拡張することなく境界処理を実現することができる。

＜実施形態2＞

実施形態1では、低域・高域ウェーブレット変換係数の演算にそれぞれ、9タップと7タップのデータを用いる9×7フィルタの場合について説明した。

【0071】

実施形態2では、L i f t i n g 係数が2のべき乗だけからなる、5タップと3タップのデータからなる5×3フィルタで、低域・高域ウェーブレット変換係数を演算する場合の構成を示す。

【0072】

L i f t i n g 係数が2のべき乗だけなので、乗算器を使う必要が無く、位取りを変更するだけで、L i f t i n g 係数の乗算ができる。

【0073】

3タップのデータから求める高域ウェーブレット変換係数H、5タップのデータから求める低域ウェーブレット変換係数Lはそれぞれ以下の式で演算する。

【0074】

$$H_{2n+1} = X_{2n+1} - 0.5 X_{2n} - 0.5 X_{2n+2} \quad (21)$$

$$L_{2n+2} = X_{2n+2} + 0.25 H_{2n+1} + 0.25 H_{2n+3} \quad (22)$$

図20に、該ウェーブレット変換処理部の構成を示す。同図において、2001はL i f t i n g 係数0.5を乗算する処理に相当する位取り変換部であり、ハードウェア上の実体は配線を1ビットずらすだけである。該構成図は水平・垂直方向いずれの処理にも共通で使えるよう演算途中のデータである中間データを蓄えるバッファを遅延器2003とした。該遅延器2003がレジスタ1段である場合は水平方向、ラインメモリである場合には垂直方向のウェーブレット変換処理部とし動作する。また、2011はL i f t i n g 係数0.25を乗算する処理に相当する位取り変換部である。

【0075】

逆変換は以下の式で演算する。

【 0 0 7 6 】

$$X_{2n+2} = L_{2n+2} - 0.25 H_{2n+1} - 0.25 H_{2n+3} \quad (23)$$

$$X_{2n+1} = H_{2n+1} + 0.5 X_{2n} + 0.5 X_{2n+2} \quad (24)$$

図 2 1 に、該ウェーブレット逆変換処理部の構成を示す。1 段目の L i f t i n g 係数 0. 5 から 0. 2 5 へ、2 段目の L i f t i n g 係数が 0. 2 5 から 0. 5 へ変わっただけで、その他は前記図 2 0 とまったく同じ構成である。

【 0 0 7 7 】

図 2 2 に、順方向と逆方向の両方のウェーブレット変換処理が可能な構成を示す。順方向変換をする時、セレクト 2 2 0 1、2 2 0 3 は端子 a 側を選択し、逆方向変換をする時、該セレクトは端子 b 側を選択するよう制御する。

【 0 0 7 8 】

上記 L i f t i n g 係数で、順方向と逆方向のウェーブレット変換処理をする場合、図 9 のようにセレクトで演算ユニット間の接続状態を切り換えることも可能ではあるが、それよりも、図 2 2 のように L i f t i n g 係数を切り換える構成としたほうが、L i f t i n g 係数乗算後の値を切り換えるセレクトを追加するだけなので、増設するセレクトの数が少ない上に加減算器を使う必要も無いのでハードウェア規模が小さくなる。

【 0 0 7 9 】

上記 3 種類の構成は、上述の図 7 にも適用できる。すなわち、図 2 3 の構成で順方向のウェーブレット変換処理ができ、図 2 4 の構成で順方向と逆方向の両方のウェーブレット変換処理ができる。

【 0 0 8 0 】

以上説明したように、実施形態 2 によれば、L i f t i n g 係数が 2 のべき乗だけからなる場合において、L i f t i n g 格子構造上の格子点データを演算する過程で生成される中間データを遅延器を用いて、この中間データを用いた演算を可能にすることで、入力されたデータストリームを先頭から 2 画素ずつペアにしてウェーブレット変換処理を行うことができる。

<実施形態 3>

実施形態 3 では、可逆（ロスレス）変換と非可逆（ロッキー）変換の切り換えが容易な構成について示す。変換処理して得られた変換係数をそのまま逆変換処理した時に元のデータを完全に復元できるような変換・逆変換を可逆変換という。画像データはデータ量が多いので、通常は、圧縮率が高くできる非可逆圧縮を用いている。しかし、診断に用いる医療用の画像など一部の画像では画像劣化の無い可逆圧縮（可逆変換）が求められる。

【 0 0 8 1 】

図 2 0 ～図 2 4 のいずれの構成においても、丸め処理器を追加することにより、通常の非可逆変換を可逆変換に変更することができる。

【 0 0 8 2 】

丸め処理を導入することにより可逆変換が実現できる理由を簡単に述べる。

【 0 0 8 3 】

通常は、実数計算による演算誤差が発生し、その誤差が累積することによって、量子化しない（量子化誤差が無い）場合でも、逆変換後に元と同じ値にならないことがしばしばある。

【 0 0 8 4 】

しかし、丸め処理により整数化を行なう場合、該丸め処理が誤差の発生源となり、丸め処理が一意に定義されていて、順方向の変換処理と逆方向の変換処理で整合がとれていれば、誤差は打ち消されて無くなるので、逆変換後に元の値に戻ることが理論的に保証されるわけである。

【 0 0 8 5 】

図 2 2 と図 2 3 に対して、丸め処理器を追加した構成を、図 2 5、図 2 6 に示す。入力データを整数データと仮定すると、この場合の丸め処理は、演算処理によって発生する小数部を整数化することである。

【 0 0 8 6 】

丸め処理の方法としては、四捨五入、切り捨て、切り上げなどいろいろあるが、図 2 5 における初段の丸め処理器 2 5 0 1、2 段目の丸め処理器 2 5 0 3 いずれも、順方向変換時は四捨五入を行なう。逆方向変換時には、初段の丸め処理器 2 5 0 1 は 0. 7 5 以上を切り上げて、0. 5 以下を切り捨て処理し、2 段目の

丸め処理器 2 5 0 3 は切り捨てを行なう。

【 0 0 8 7 】

図 2 6 における初段の丸め処理器 2 6 0 1 は順方向変換時に切り捨てを行ない、逆方向変換時に四捨五入を行なう。2 段目の丸め処理器 2 6 0 3 は順方向変換時に四捨五入を行ない、逆方向変換時に切り捨て処理を行なう。

【 0 0 8 8 】

よって、各丸め処理器は変換処理の種類に対応して、丸め処理の内容を切り換えなければならないため、その切り換えのための制御信号を入力している。

【 0 0 8 9 】

上記丸め処理器の前段の加算器や減算器でオフセットを加算することで、該丸め処理器での処理を切り捨てだけにもできる。ここで言う切り捨てとは小数部の切り捨てのことを言っている。

【 0 0 9 0 】

図 2 5、図 2 6 に対してオフセットを加算する場合の構成を図 2 7 と図 2 8 に示す。同図における丸め処理器は上述のごとく小数部の切り捨てだけをする。加算するオフセット値は変換の種類（順方向・逆方向）によって異なり、図 2 7 におけるオフセット生成器 2 7 0 1 は順方向変換時には 0. 5 を、逆方向変換時には 0. 2 5 を出力し、もう 1 つのオフセット生成器 2 7 0 3 は順方向変換時には 0. 5 を、逆方向変換時には 0 を出力する。

【 0 0 9 1 】

図 2 8 におけるオフセット生成器 2 8 0 1 は順方向変換時には 0 を、逆方向変換時には 2 を出力し、もう 1 つのオフセット生成器 2 8 0 3 は順方向変換時には 2 を、逆方向変換時には 0 を出力する。

【 0 0 9 2 】

いずれのオフセット生成器も変換処理の種類に対応して、出力するオフセット値を切り換えなければならないため、その切り換えのための制御信号を入力している。逆に、丸め処理は一律に小数部を切り捨てるだけであるから、制御信号を入力する必要がない。この場合の丸め処理器の物理的な実体は無く、小数部の信号の配線が断線状態になっている。

【 0 0 9 3 】

小数部の切り捨てで丸め処理を行なう図 2 7、図 2 8 の構成から、次のような新たな構成も考えられる。

【 0 0 9 4 】

丸め処理器を、該小数部の信号を制御信号でマスクする構成に変えることにより、該制御信号で丸め処理の ON / OFF 制御ができ、可逆変換と非可逆変換の切り換えが容易に行なえるようになる。もちろん、丸め処理の ON / OFF だけでなく、加算器・減算器に入力するオフセットも連動して ON / OFF する必要がある。

【 0 0 9 5 】

図 2 9、図 3 0 に示す構成は、図 2 7、図 2 8 の構成に対して、可逆 / 非可逆変換を切り換える制御信号をあらたに設け、該制御信号によって、可逆変換時には丸め処理器において小数部をマスクし、非可逆変換時にはオフセット生成器においてオフセット出力をマスクする。

【 0 0 9 6 】

上述した制御により、オフセットの加算と小数部の切り捨てを行なう可逆変換処理と、オフセットの加算も小数部の切り捨ても無い非可逆変換処理の切り換えが容易に行なえるウェーブレット変換装置が構成できるようになった。

【 0 0 9 7 】

また、図 2 5、図 2 6 において、可逆 / 非可逆変換を切り換える制御信号により丸め処理器をパスするようにセレクタを設けることで、該可逆 / 非可逆変換を切り換えることも可能である。

【 0 0 9 8 】

以上説明したように、実施形態 3 によれば、実施形態 2 で説明した効果に加えて、実施形態 2 で説明した構成に対し丸め処理器、オフセット生成器を追加した簡単な回路構成で、ウェーブレット変換における可逆変換と非可逆変換の切り換えを容易に行うことができる。

< 実施形態 4 >

これまでの実施形態は、水平方向のウェーブレット変換もしくは垂直方向のウ

ウェーブレット変換のどちらか一方の処理を行なうものであった。実施形態 4 では水平・垂直の 2 次元ウェーブレット変換処理を行なう構成について示す。

【0099】

前記 L i f t i n g 演算ユニット、あるいは、格子点演算ユニットを用いた 2 次元ウェーブレット変換処理器は図 3 1 のような構成となる。該ウェーブレット変換処理器は前述の 2 のべき乗の L i f t i n g 係数を用いた 5×3 フィルタの処理をするものである。

【0100】

同図において、3 1 0 1 は、順方向変換時のデータを入力する端子である。3 1 0 3 は、逆方向変換時のデータを入力する端子である。3 1 1 1 と 3 1 1 3 は、垂直方向のウェーブレット変換処理を行なう演算ユニットである。3 1 2 1 と 3 1 2 3 は、水平方向のウェーブレット変換処理を行なう演算ユニットである。3 1 1 5 と 3 1 2 5 は、変換モード（順方向変換・逆方向変換）に応じて上記演算ユニットへの入力を切り換えるセレクタである。3 1 1 7 と 3 1 2 7 は、水平または垂直ウェーブレット変換されたデータを 2×2 単位で回転して、次の段のウェーブレット変換処理部に供給するデータ回転部である。3 1 3 0 は、順方向または逆方向 2 次元ウェーブレット変換されたデータを一時的に蓄えて外部のメモリ等へ出力するためのバッファである。

【0101】

順方向変換時には、端子 3 1 0 1 から入力されたデータがセレクタ 3 1 1 5 を経由して演算ユニット 3 1 1 1 に供給される。該演算ユニット 3 1 1 1 と次の演算ユニット 3 1 1 3 とで垂直方向のウェーブレット変換処理を行ない、変換結果をデータ回転部 3 1 1 7 へ送る。

【0102】

ここで、2 つのデータ回転部 3 1 1 7、3 1 2 7 の簡単な説明を行なう。データ回転部 3 1 1 7、3 1 2 7 は図 3 2 に示すような構成で、2 つ並列に入力されるデータを 2 サイクル周期で 2×2 単位の回転を行ない、次段のセレクタへ送る。

【0103】

図 3 2 において、データ入力端子 3 2 0 1 と 3 2 0 3 から並列に入力されたデータは、それぞれレジスタ 3 2 1 1 ~ 3 2 1 5 にて、1 サイクル毎にシフトしながら格納される。

【0 1 0 4】

m サイクル目と m + 1 サイクル目に端子 3 2 0 1 に入力されたデータは、m + 2 サイクル目にセレクタ 3 2 2 1 と 3 2 2 3 を経由して端子 3 2 3 1 と 3 2 3 3 から出力される。

【0 1 0 5】

端子 3 2 0 3 へ上記データと同時に並列入力されたデータは、m + 3 サイクル目に端子 3 2 3 1 と 3 2 3 3 から出力される。

【0 1 0 6】

m + 2 サイクル目と m + 3 サイクル目に入力された 2×2 のデータは m + 4、m + 5 サイクル目に出力される。

【0 1 0 7】

データ回転部 3 1 1 7 では、垂直方向 2 サンプルの並列データが水平方向 2 サンプルの並列データに変換され、データ回転部 3 1 2 7 では、水平方向 2 サンプルの並列データが垂直方向 2 サンプルの並列データに変換される。

【0 1 0 8】

データ回転部 3 1 1 7 によって変換された、水平方向 2 サンプルの並列データはセレクタ 3 1 2 5 を経由して、演算ユニット 3 1 2 1 に入力される。演算ユニット 3 1 2 1、3 1 2 3 からなる水平ウェーブレット変換部は該入力データを変換処理し、変換結果を演算ユニット 3 1 2 3 からバッファ 3 1 3 0 へ出力する。上記水平ウェーブレット変換部は、垂直方向の低域ウェーブレット変換係数と高域ウェーブレット変換係数を交互に処理するため、前記演算ユニット 3 1 2 1、3 1 2 3 内の各遅延器は 2 段のレジスタからなる。

【0 1 0 9】

逆方向変換時には、端子 3 1 0 3 から入力されたデータがセレクタ 3 1 2 5 を経由して演算ユニット 3 1 2 1 に供給される。不図示の制御信号により、逆方向ウェーブレット変換処理モードとなった演算ユニット 3 1 2 1 と次の演算ユニッ

ト 3 1 2 3 とで水平方向の逆ウェーブレット変換処理を行ない、変換結果をデータ回転部 3 1 2 7 へ送る。

【 0 1 1 0 】

逆方向変換時の水平ウェーブレット変換部におけるデータの処理順序は、順方向変換時に出力するデータをそのまま逆に処理すればよいが、この順序は通常の 1 ライン単位の処理とは異なるので、少し補足説明する。

【 0 1 1 1 】

実施形態 4 の構成では 2 ライン単位で処理をする。但し、2 ラインを同時に並列処理することはできないので、ライン交互に水平 2 サンプルを入力して、逆方向の水平ウェーブレット変換処理を行なう。演算ユニット内の各遅延器はレジスタ 2 段で構成されているので、2 ラインを交互に処理することが可能である。逆変換処理した結果も当然 2 ライン交互に出力される。それを 2 × 2 のデータ回転処理部 3 1 2 7 で 2 ライン並列のデータに変換することで、垂直方向のウェーブレット変換処理部に都合のよいデータを送ることができる。

【 0 1 1 2 】

その後、セクタ 3 1 1 5 で上記データを選択して演算ユニット 3 1 1 1 に入力すれば、垂直方向の逆ウェーブレット変換処理が行なわれ、変換結果が演算ユニット 3 1 1 3 からバッファ 3 1 3 0 へ出力される。

【 0 1 1 3 】

9 × 7 フィルタで処理する場合には、垂直処理部の演算ユニット 3 1 1 1、3 1 1 3 並びに水平処理部の演算ユニット 3 1 2 1、3 1 2 3 を、それぞれ図 9 に示す構成に置き換えればよい。

【 0 1 1 4 】

以上説明したように、実施形態 4 によれば、実施形態 1 ～実施形態 3 で説明した構成を利用して、簡単な回路構成で、入力されたデータストリームを先頭から 2 ラインずつ 2 次元ウェーブレット変換処理を行うことができる。

【 0 1 1 5 】

尚、上述した本発明で実現されるハードウェア構成は、専用ハードウェア基板あるいは専用ハードウェアチップとしてパーソナルコンピュータ等の端末に実装

され、その端末が有するCPU等の制御部が上述した制御あるいは制御信号の生成を行うことにより、該ハードウェア構成による処理が実行される。

【0116】

また、本発明で実現されるハードウェア構成による処理をソフトウェアとして端末の記憶装置に記憶しておき、このソフトウェアを端末に搭載されるCPUによって実行させることも可能である。このソフトウェアが実行する処理について、以下、図32に示す。

【0117】

図32は本発明で実行される処理を示すフローチャートである。

【0118】

まず、ステップS101で、データを入力する。次に、ステップS102で、上述の(9)～(12)式を用いて、L i f t i n g格子構造上の格子点データを演算する途中のデータである中間データを生成する。次に、ステップS103で、生成した中間データを保持する。次に、ステップS104で、上述の(13)～(16)式を用いて、保持した中間データから格子点データを生成する。

【0119】

ステップS105で、処理対象のデータが存在するか否かを判定する。処理対象のデータが存在する場合(ステップS105でYES)、ステップS106に進み、次の処理対象のデータを入力し、ステップS102に戻る。一方、処理対象のデータが存在しない場合(ステップS105でNO)、処理を終了する。

【0120】

尚、本発明は、複数の機器(例えばホストコンピュータ、インタフェース機器、リーダ、プリンタなど)から構成されるシステムに適用しても、一つの機器からなる装置(例えば、複写機、ファクシミリ装置など)に適用してもよい。

【0121】

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ(またはCPUやMPU)が記憶媒体に格納されたプログラムコードを読出し実行することによっても、達成されることは言

うまでもない。

【0 1 2 2】

この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0 1 2 3】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R/RW、DVD-ROM/RAM、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0 1 2 4】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0 1 2 5】

更に、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0 1 2 6】

本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明した図3に示すフローチャートに対応するプログラムコードが格納されることになる。

【0 1 2 7】

【発明の効果】

以上説明したように、本発明によれば、効率的にウェーブレット変換処理を実

行できるデータ処理装置及びその制御方法、データ処理方法、コンピュータ可読メモリを提供できる。

【図面の簡単な説明】

【図 1】

順変換の L i f t i n g S c h e m e を示す図である。

【図 2】

逆変換の L i f t i n g S c h e m e を示す図である。

【図 3】

DWT（順方向ウェーブレット変換）の L i f t i n g 格子構造を示す図である。

【図 4】

IDWT（逆方向ウェーブレット変換）の L i f t i n g 格子構造を示す図である。

【図 5】

本発明の内容を L i f t i n g 格子構造で表現した図である。

【図 6】

実施形態 1 の構成を示す図である。

【図 7】

格子点データ演算ユニットの構成を示す図である。

【図 8】

実施形態 1 の応用例である IDWT 変換装置の構成を示す図である。

【図 9】

実施形態 1 の応用例である DWT / IDWT 変換装置の構成を示す図である。

【図 1 0】

実施形態 1 の応用例である DWT / IDWT 変換装置の構成を示す図である。

【図 1 1】

始端境界処理が可能な L i f t i n g 演算ユニットの構成を示す図である。

【図 1 2】

終端境界処理が可能な L i f t i n g 演算ユニットの構成を示す図である。

【図 1 3】

始端と終端の両方の境界処理が可能な L i f t i n g 演算ユニットの構成を示す図である。

【図 1 4】

L i f t i n g 格子構造中の境界部のデータを示す図である。

【図 1 5】

L i f t i n g 格子構造中の境界部のデータを示す図である。

【図 1 6】

図 1 4 の境界部のデータの処理が可能な D W T 変換器を示す図である。

【図 1 7】

図 1 4 と図 1 5 の両方の境界部のデータの処理が可能な D W T 変換器を示す図である。

【図 1 8】

D W T / I D W T が可能な L i f t i n g 演算ユニットの他の構成を示す図である。

【図 1 9】

D W T / I D W T が可能な格子点演算ユニットの他の構成を示す図である。

【図 2 0】

5 × 3 フィルタの D W T 変換処理部の構成を示す図である。

【図 2 1】

5 × 3 フィルタの I D W T 変換処理部の構成を示す図である。

【図 2 2】

5 × 3 フィルタの D W T / I D W T 変換処理部の構成を示す図である。

【図 2 3】

5 × 3 フィルタの D W T 変換処理部の他の構成を示す図である。

【図 2 4】

5 × 3 フィルタの D W T / I D W T 変換処理部の他の構成を示す図である。

【図 2 5】

5 × 3 フィルタの可逆 D W T / I D W T 変換処理部の構成を示す図である。

【図 2 6】

5×3 フィルタの可逆 DWT / I DWT 変換処理部の他の構成を示す図である。

【図 2 7】

5×3 フィルタの可逆 DWT / I DWT 変換処理部の他の構成を示す図である。

【図 2 8】

5×3 フィルタの可逆 DWT / I DWT 変換処理部の他の構成を示す図である。

【図 2 9】

5×3 フィルタの可逆 / 非可逆・DWT / I DWT 変換処理部の構成を示す図である。

【図 3 0】

5×3 フィルタの可逆 / 非可逆・DWT / I DWT 変換処理部の他の構成を示す図である。

【図 3 1】

5×3 フィルタの 2 次元 DWT / I DWT 変換装置の構成を示す図である。

【図 3 2】

データ回転部の構成を示す図である。

【図 3 3】

本発明で実行される処理を示すフローチャートである。

【符号の説明】

6 0 1、6 0 3、6 0 5、6 0 7 端子

6 1 1 乗算器

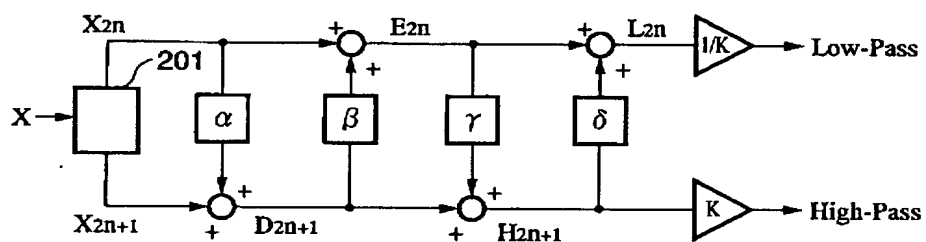
6 1 3 レジスタ

6 1 5、6 1 7 加算器

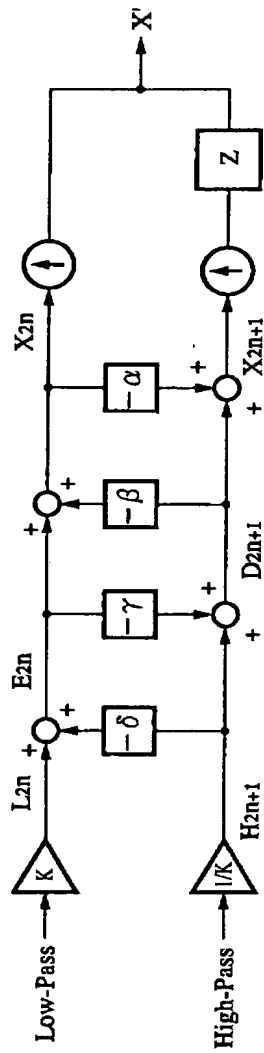
6 2 1 ~ 6 2 4 L i f t i n g 演算ユニット

【書類名】 図面

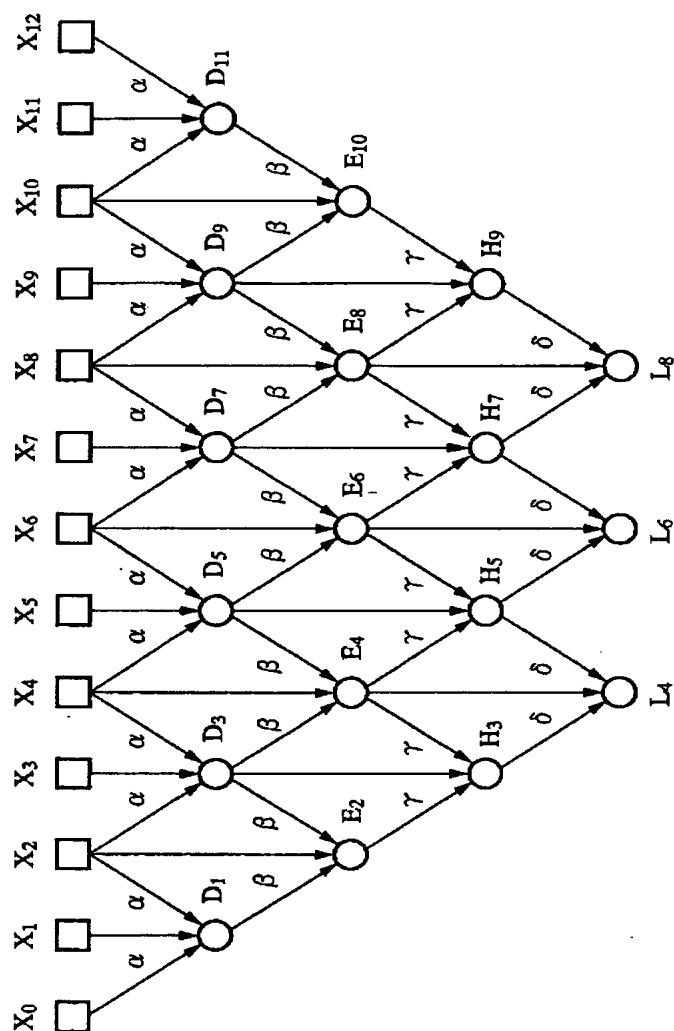
【図 1】



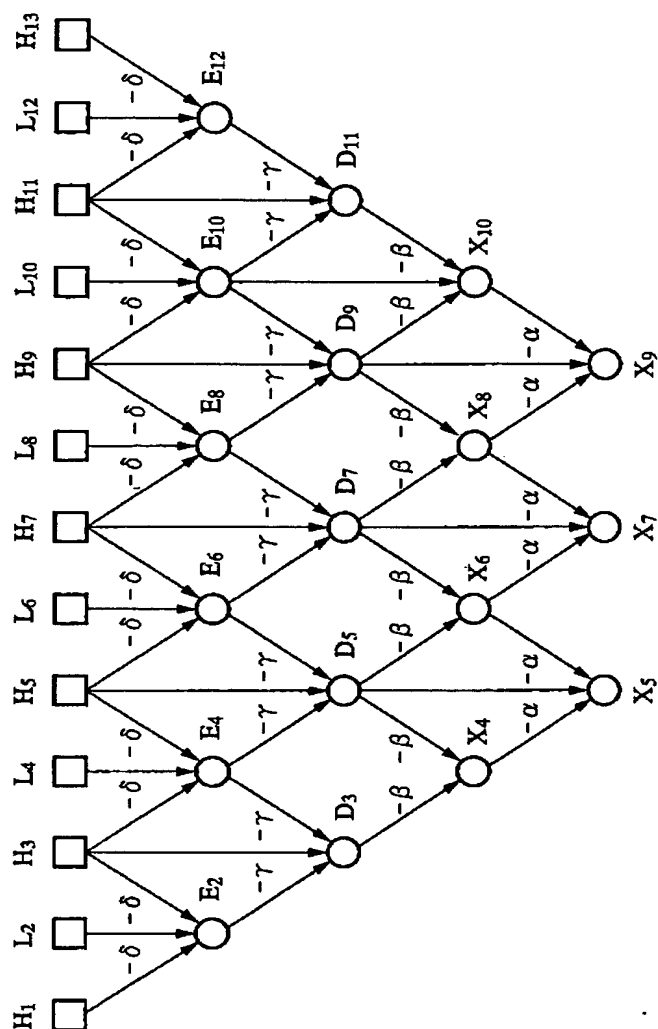
【図 2】



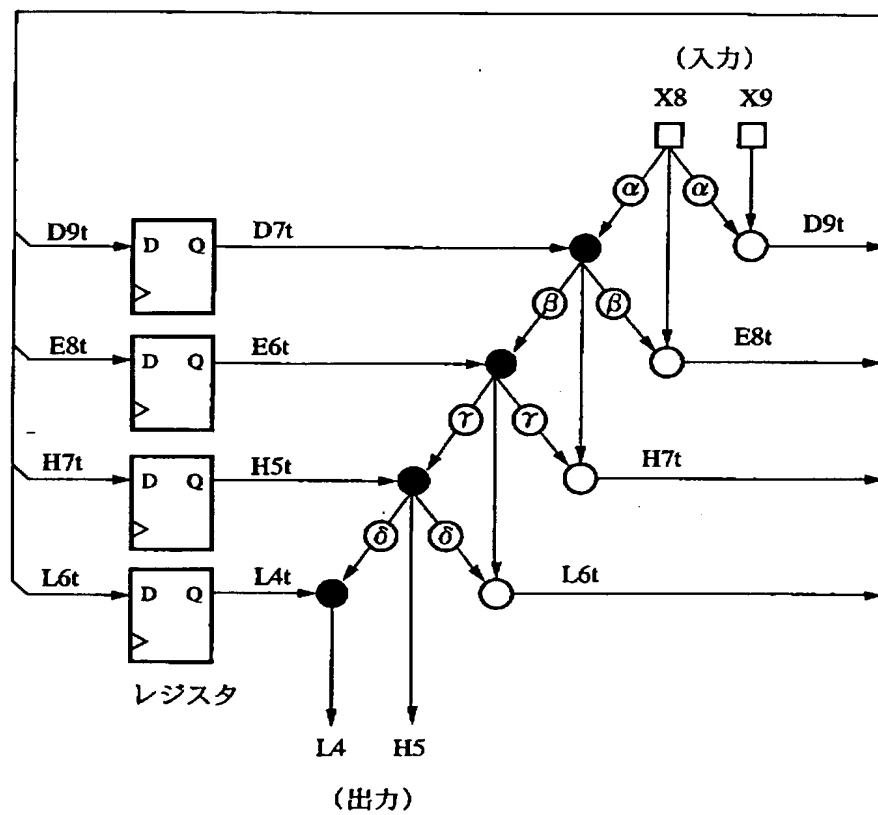
【図3】



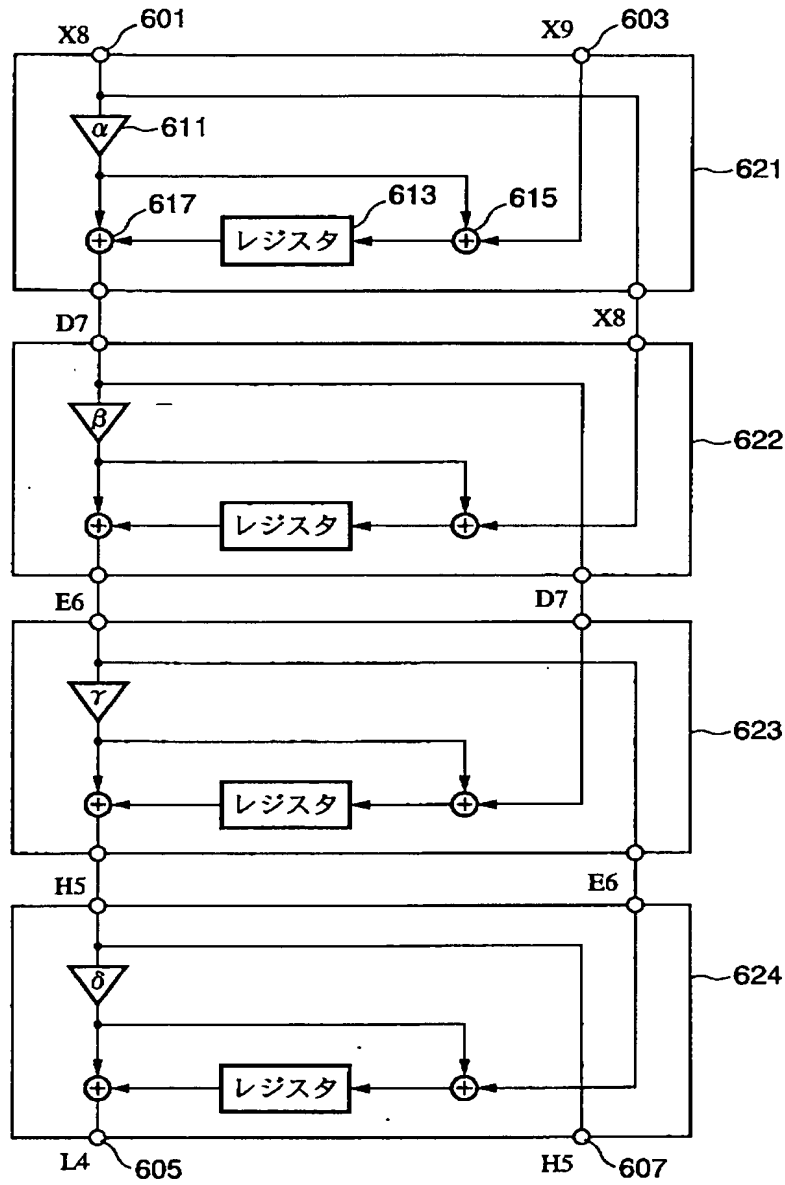
【図 4】



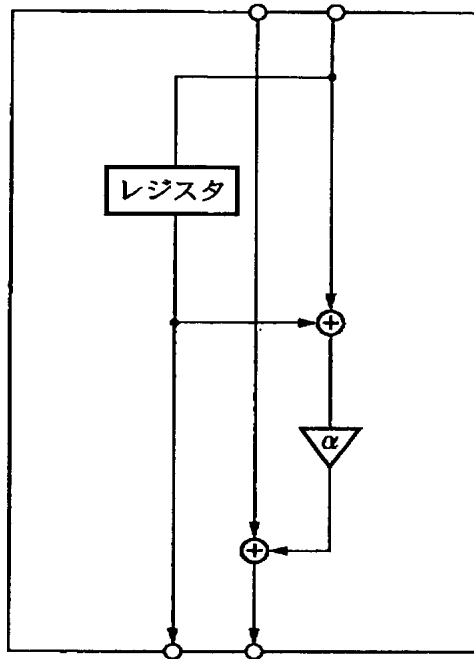
【図 5】



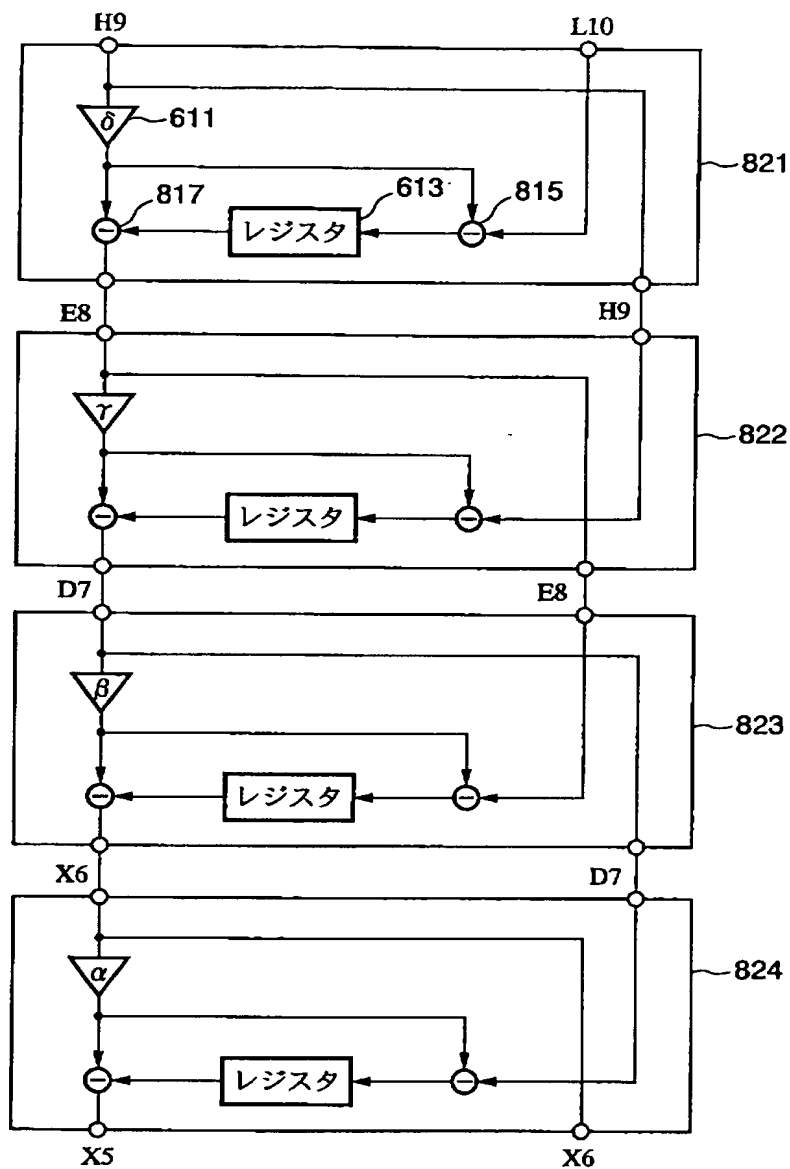
【図 6】



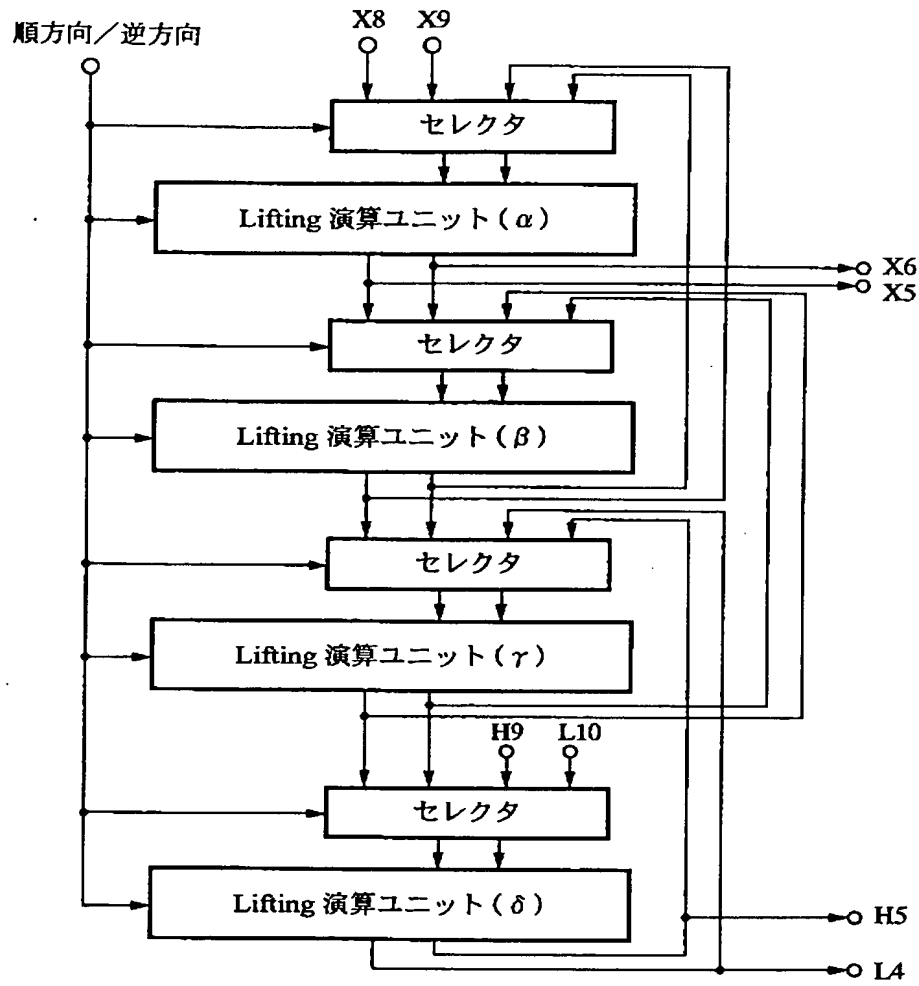
【図 7】



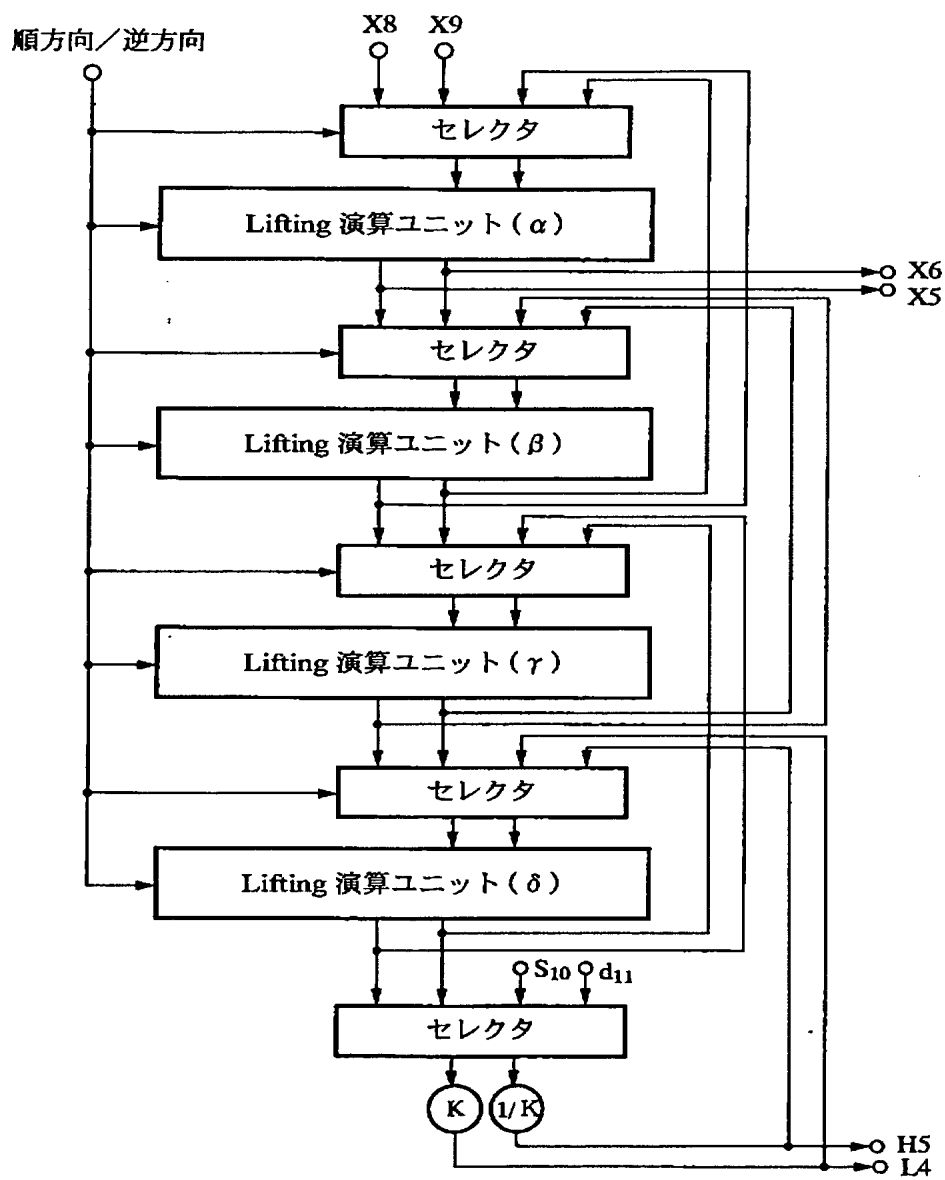
【図 8】



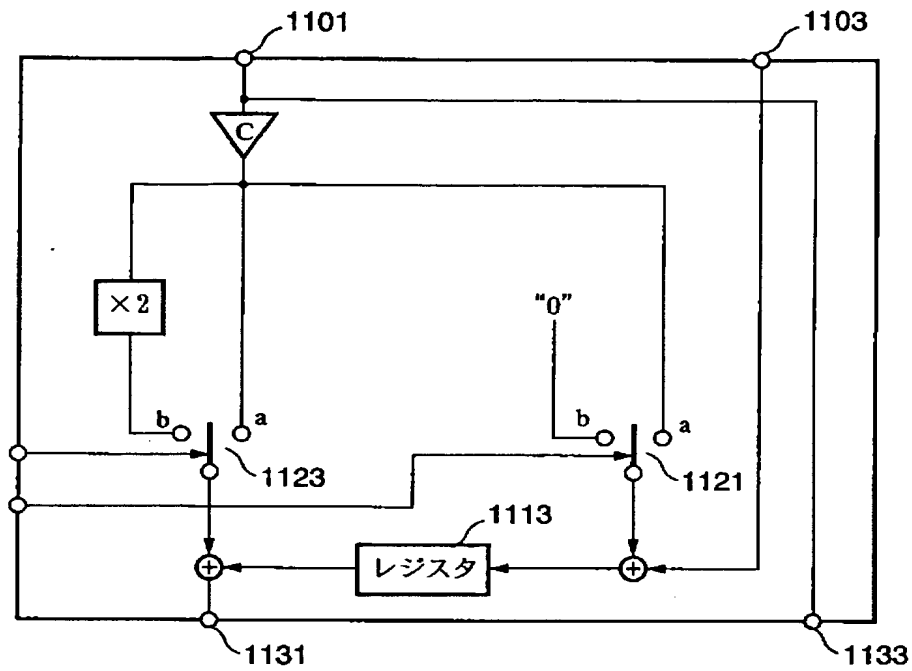
【図 9】



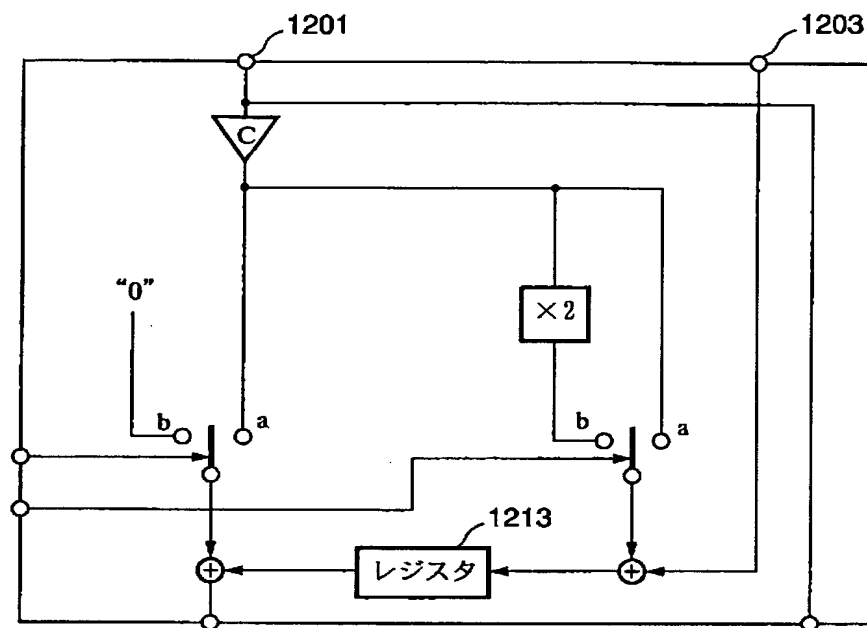
【図 10】



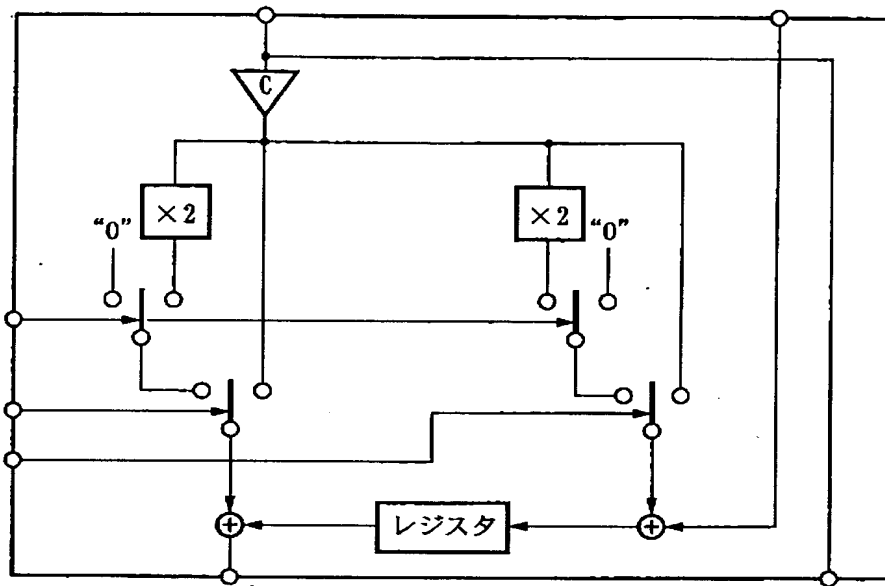
【図 1 1】



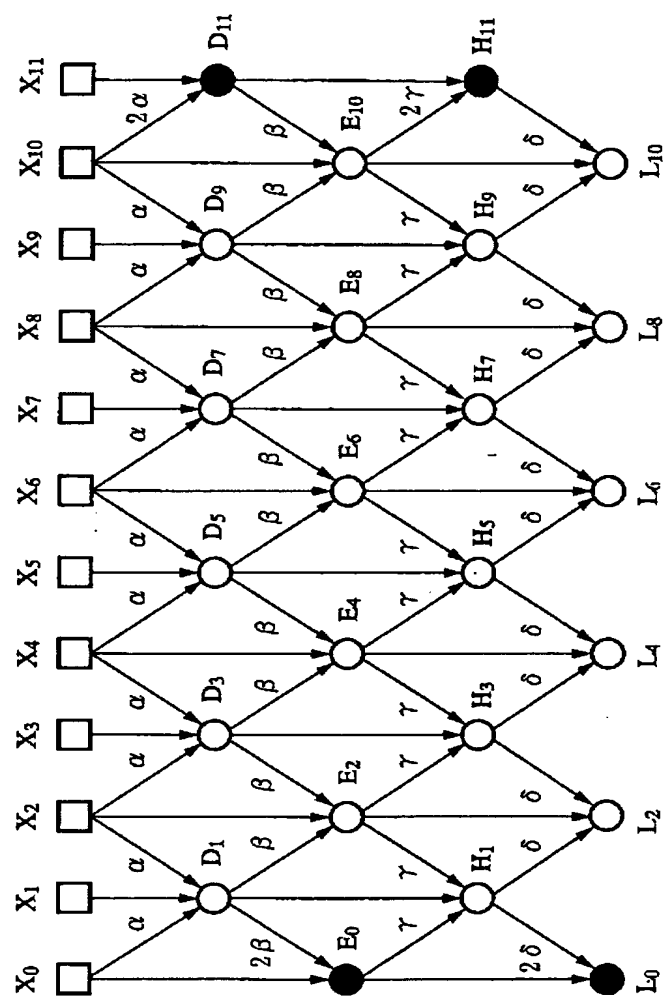
【図 1 2】



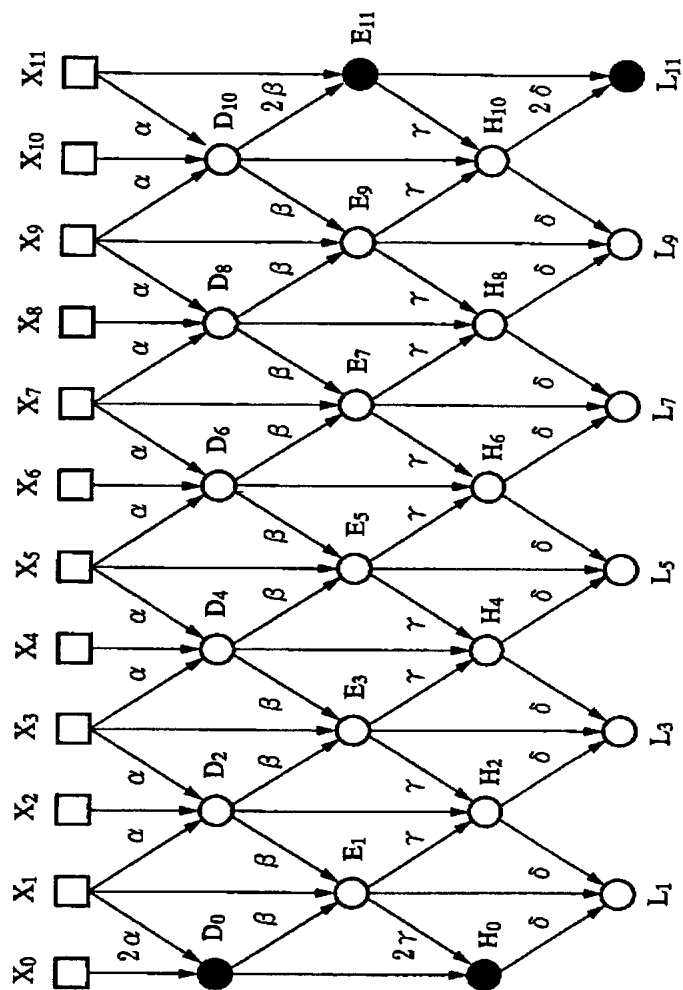
【図 1 3】



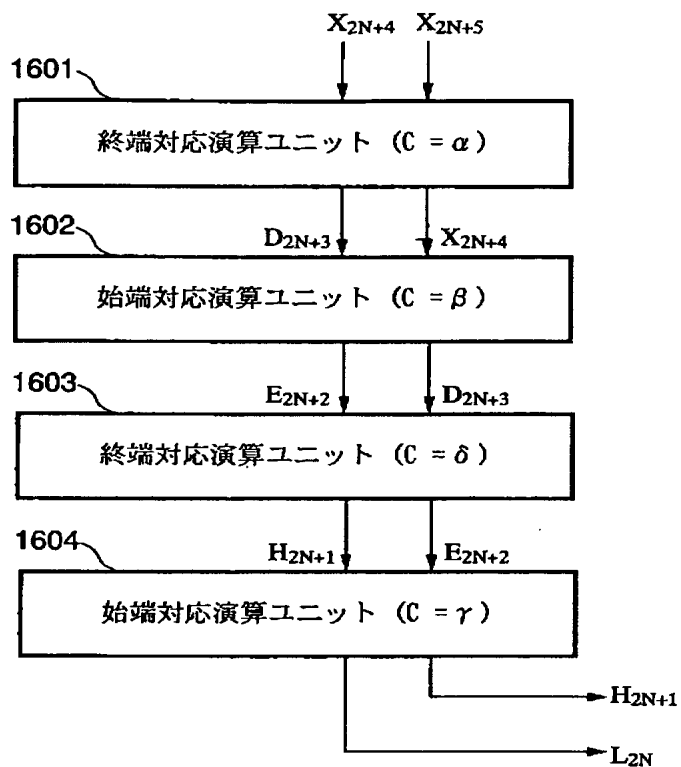
【図 14】



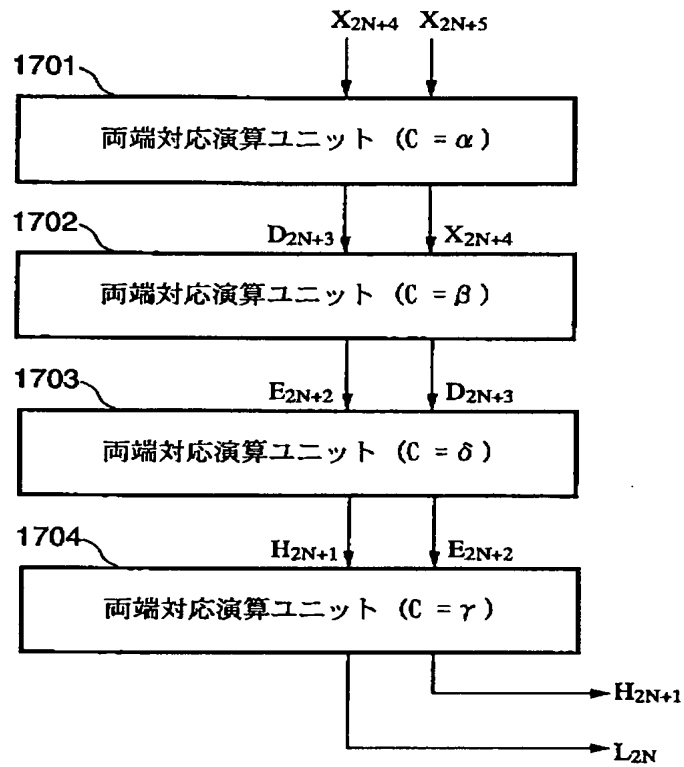
【図15】



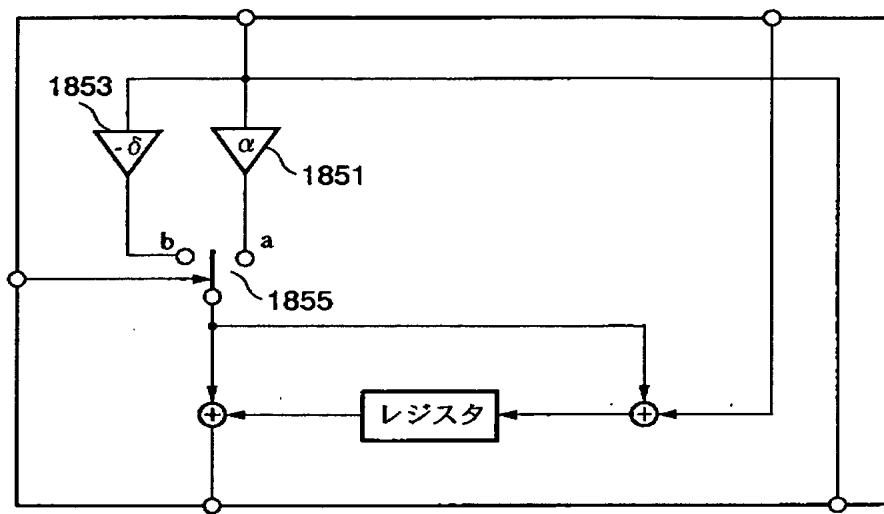
【図 1 6】



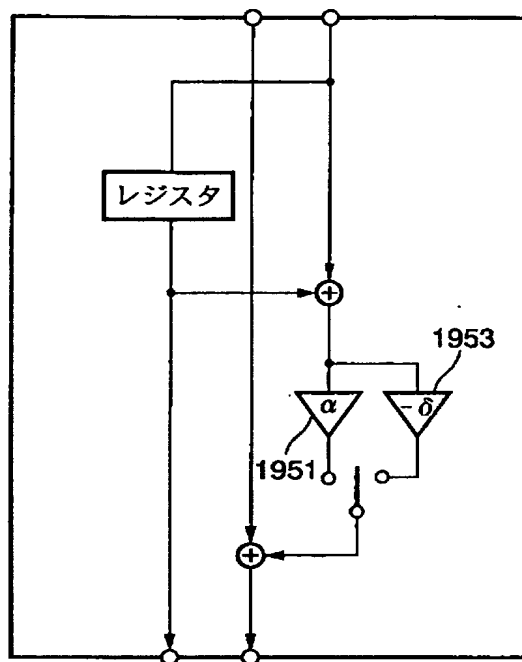
【図 1 7】



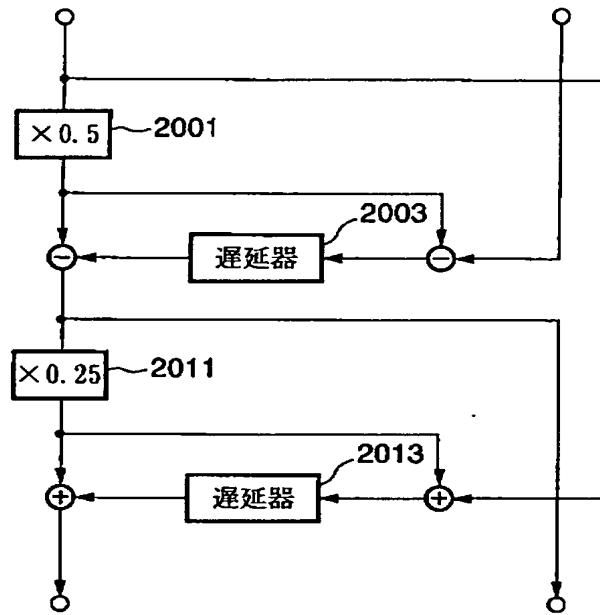
【図 1 8】



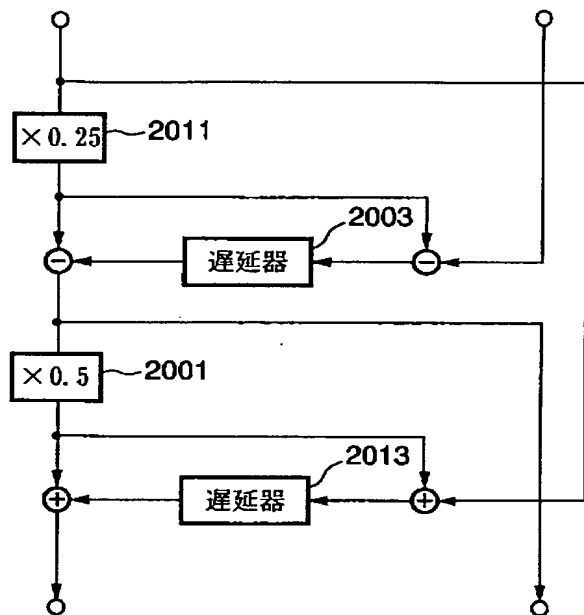
【図 1 9】



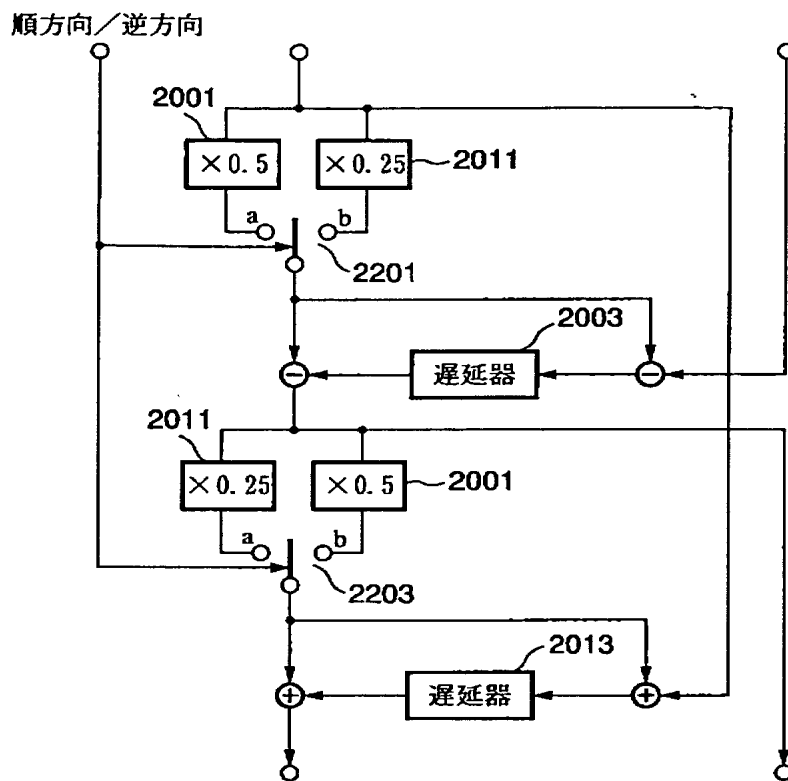
【図 2 0】



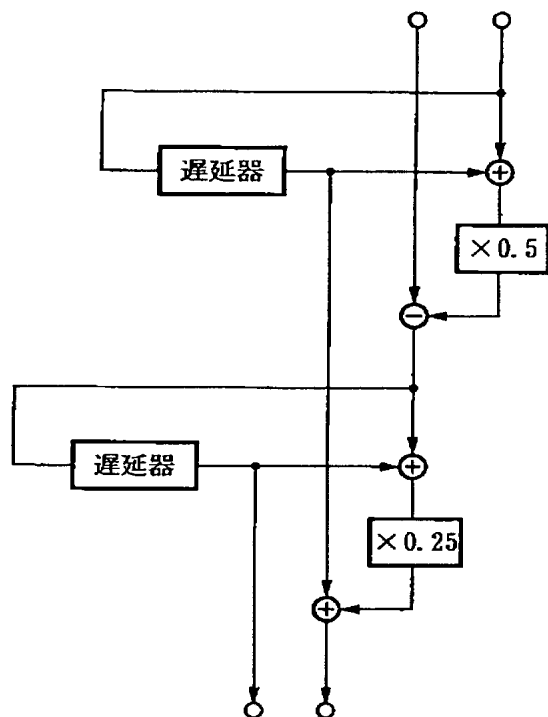
【図 2 1】



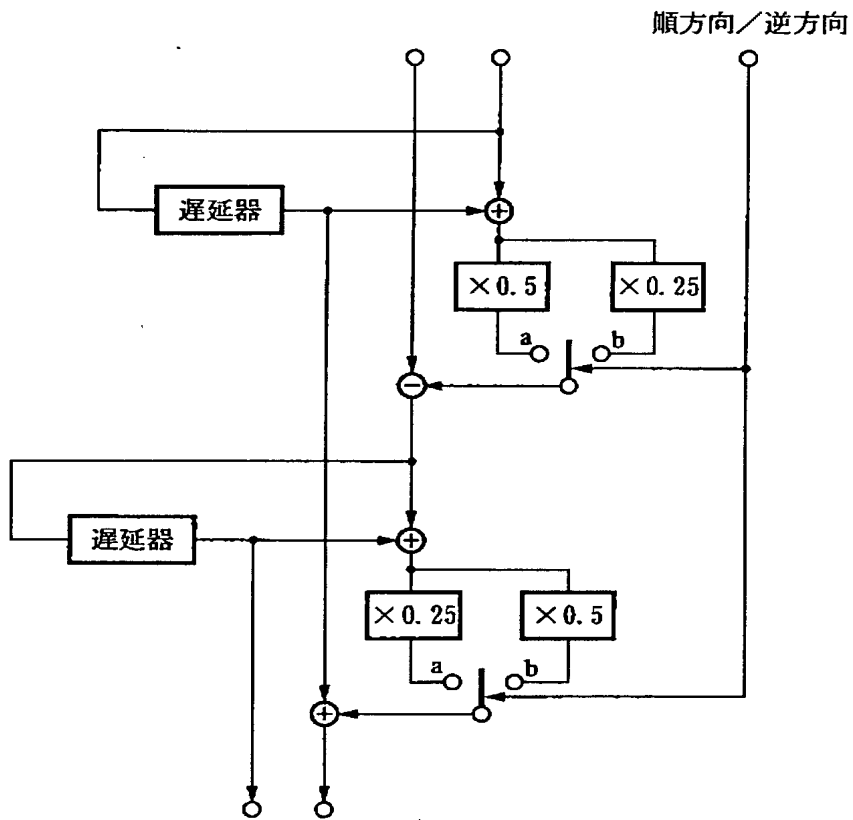
【図 2 2】



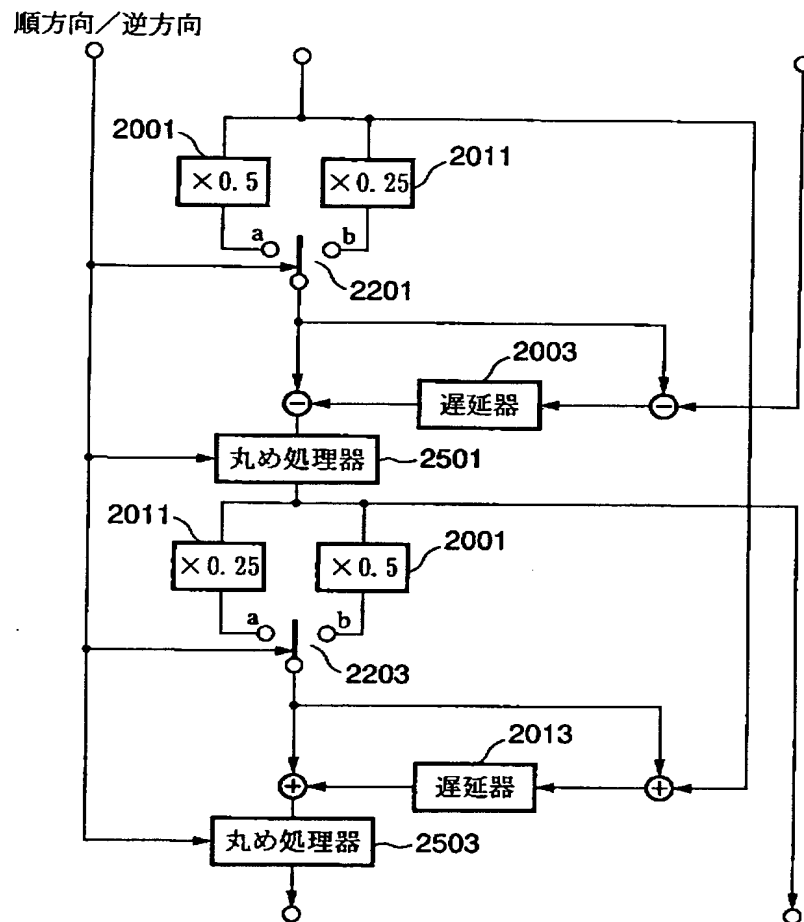
【図 2 3】



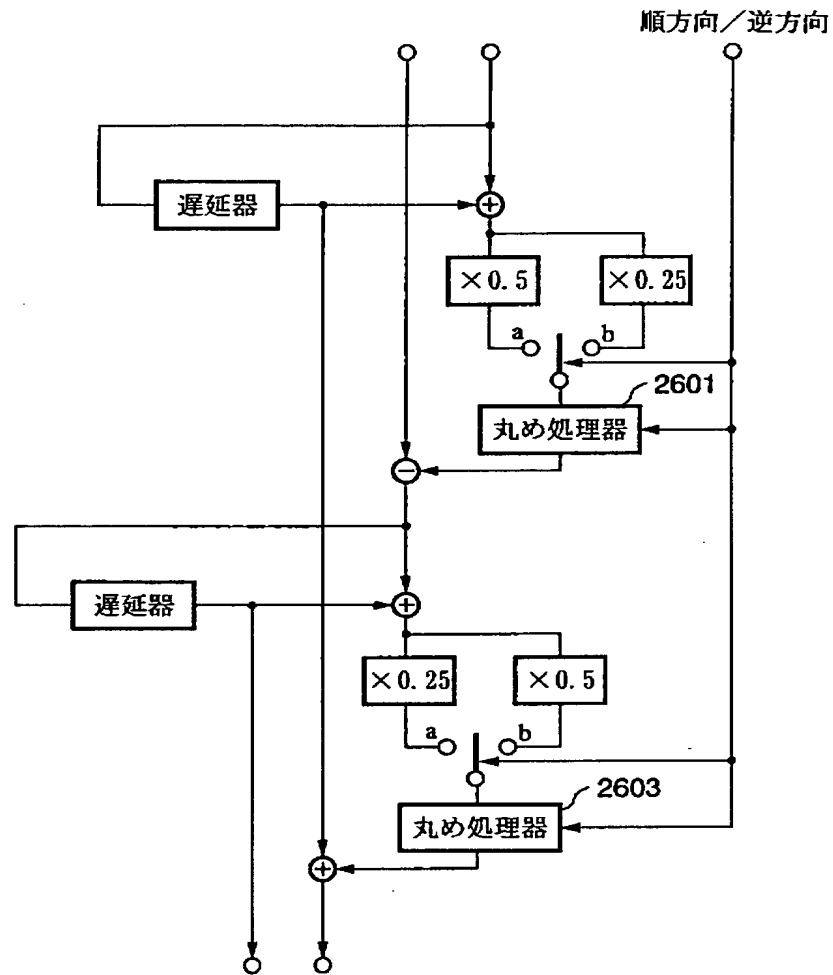
【図 2 4】



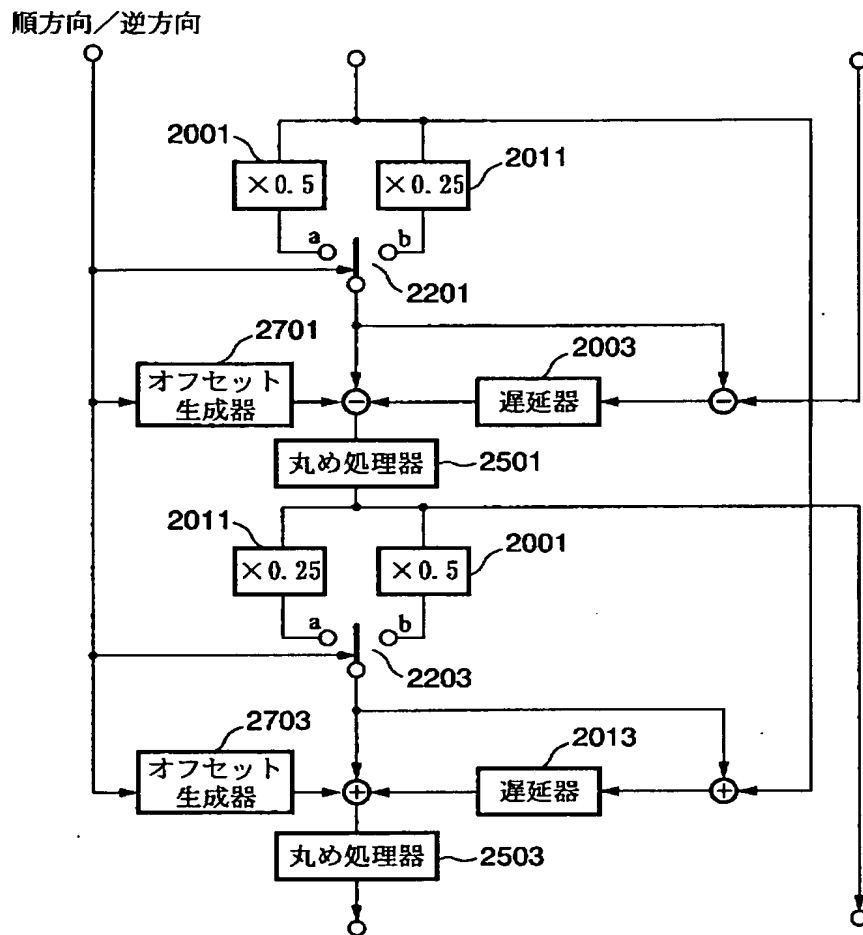
【図 2 5】



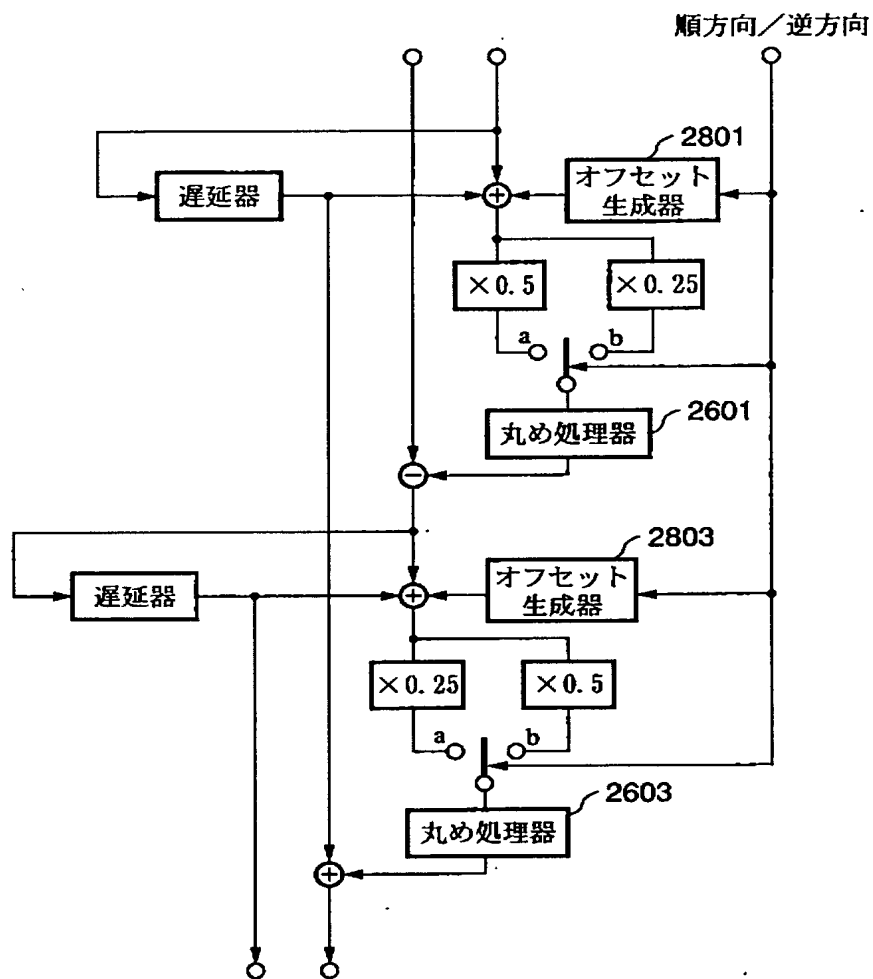
【図 2 6】



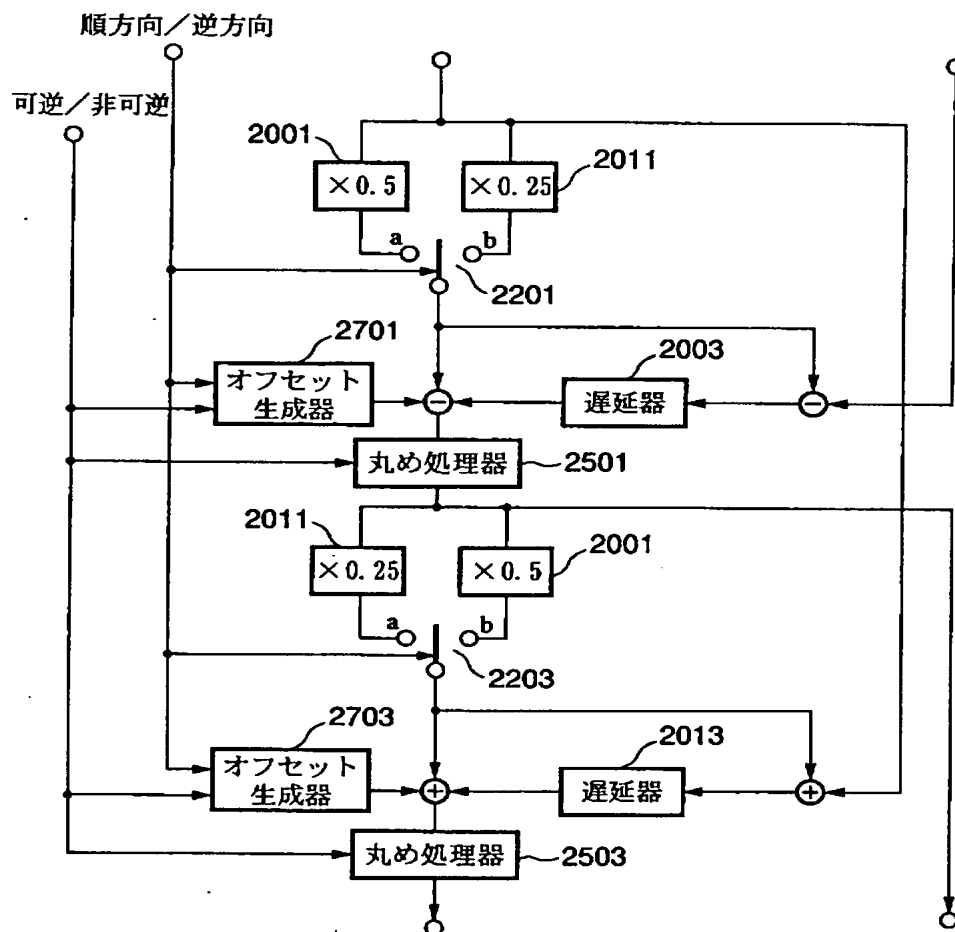
【図 2 7】



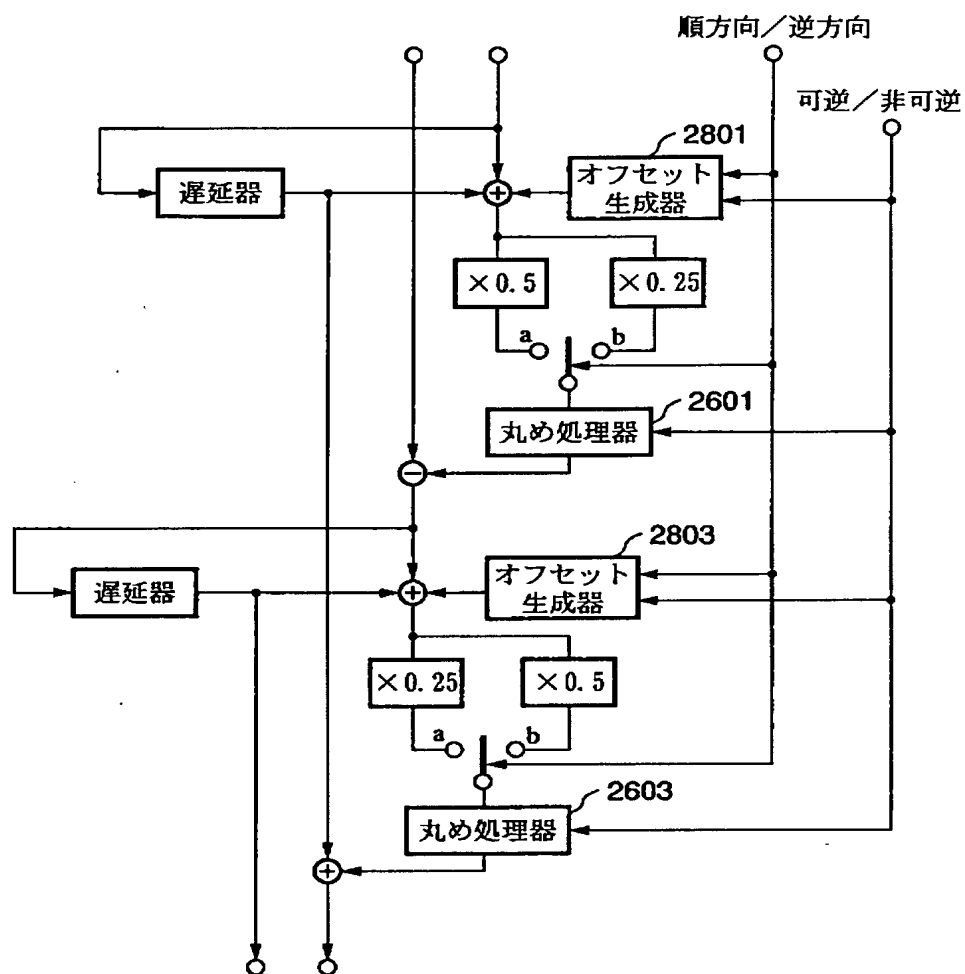
【図 2 8】



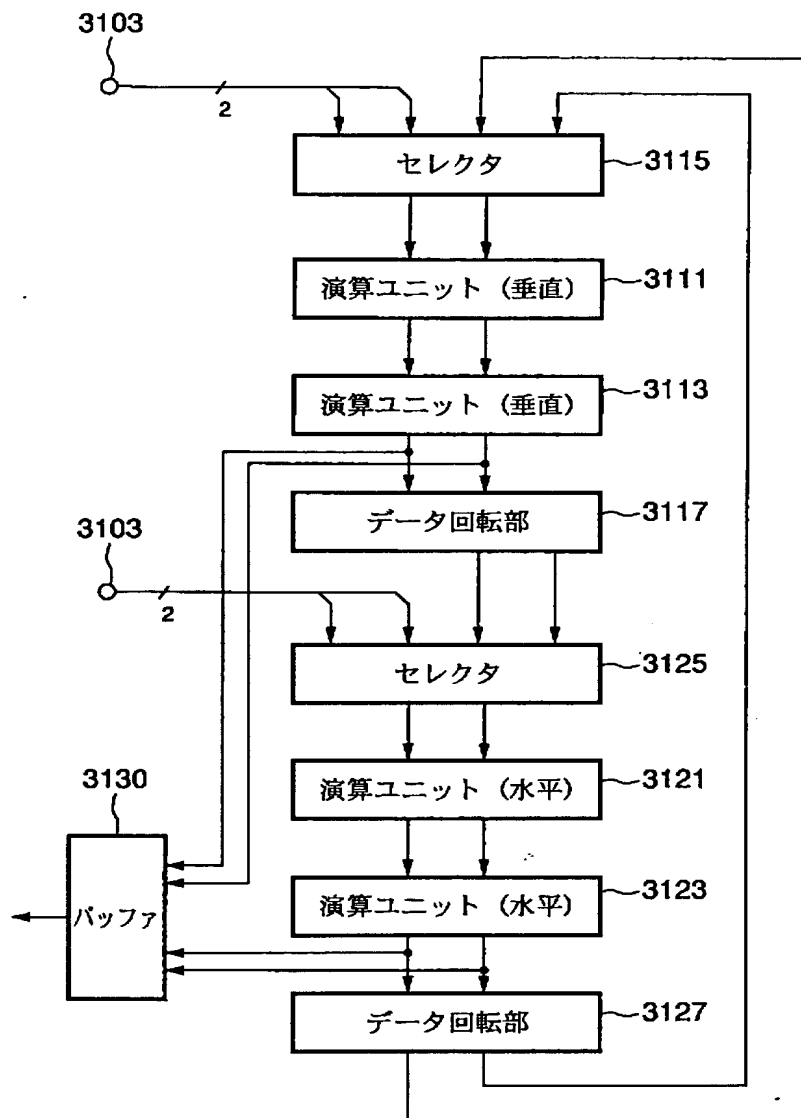
【図 2 9】



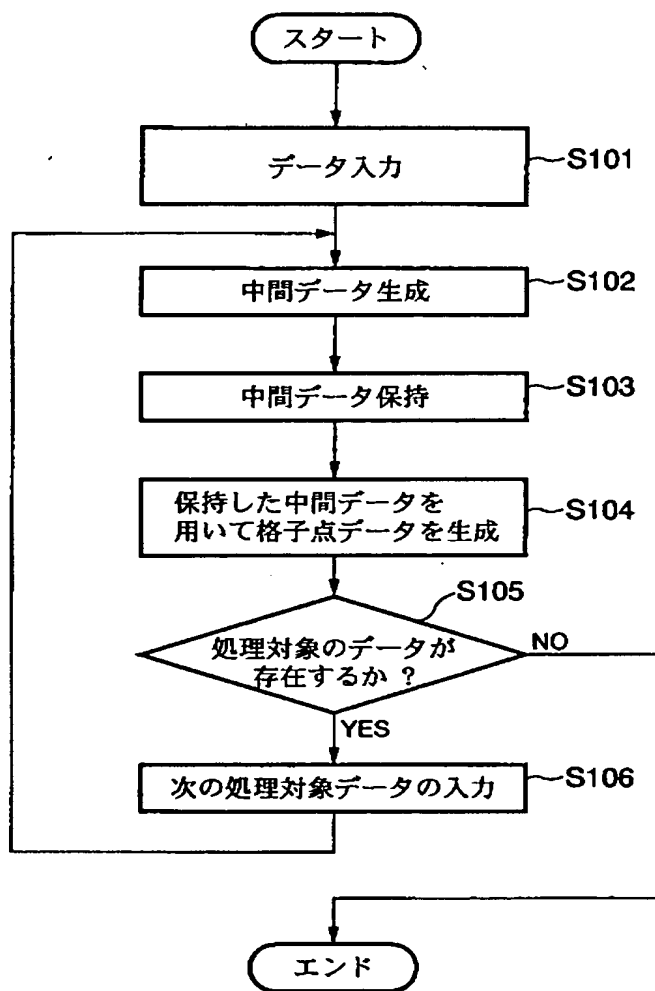
【図 3 0】



【図 3 1】



【図 3 3】



【書類名】 要約書

【要約】

【課題】 効率的にウェーブレット変換処理を実行できるデータ処理装置及びその制御方法、データ処理方法、コンピュータ可読メモリを提供する。

【解決手段】 データを処理するデータ処理装置であって、

データを保持するレジスタ 6 1 3 と、複数の加算器 6 1 5、6 1 7 と、乗算器 6 1 1 を有する演算ユニット 6 2 1 とを備え、演算ユニット 6 2 1 は、該乗算器の乗算係数が異なる演算ユニット群 6 2 2 ～ 6 2 4 と従属に接続した構成からなり、該演算ユニット群内の前記各構成要素は、 $2n+1$ タップと $2n-1$ タップのデータを入力して、ウェーブレット変換係数の演算を行うことが可能になるように構成されている。

【選択図】 図 6

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都大田区下丸子3丁目30番2号

氏 名 キヤノン株式会社



Creation date: 12-13-2004
Indexing Officer: AVU - ANHTRAM VU
Team: OIPEBackFileIndexing
Dossier: 09982916

Legal Date: 09-14-2004

No.	Doccode	Number of pages
1	CTRS	5

Total number of pages: 5

Remarks:

Order of re-scan issued on